Modelos de machine learning aplicado al cálculo de probabilidad de refacturas de clientes para empresa ESVAL S.A.

Joaquín Farias, *Practicante.*

Resumen—En este informe se expondrá el trabajo de dos meses donde se estudió el problema de las refacturas a clientes dentro de la empresa ESVAL S.A. (que incluye a la empresa Aguas del Valle) como mecanismo de determinación del error de lectura en terreno como lo que corresponde al análisis y su impacto en el error del consumo cobrado a los clientes. La práctica comienza con un análisis exploratorio de los datos para luego crear modelos de *Machine Learning* que tuvieron buenos resultados en la clasificación de los clientes por posible refactura. Además, se analizaron los tiempos de entrenamiento para una implementación con reentrenamiento.

Palabras clave— *Machine Learning*, distribución del agua, refacturas, aprendizaje supervisado, clasificación.

1 Introducción

Dentro de la empresa ESVAL S.A. se deben hacer los cobros respectivos a todos los clientes. Estos cobros se calculan a partir de lecturas hechas por personal de la empresa que debe leer, de manera manual, cada medidor de agua de cada cliente. Este proceso de lectura puede tener dificultades para llevarse a cabo de manera satisfactoria, ya sea por negligencia de los clientes al tener una accesibilidad reducida al medidor de agua, lo que dificulta el actuar del personal de la empresa, o bien, debido a negligencia de los lectores de la empresa.

El error el cálculo de los consumos dicho anteriormente intenta disminuir con un análisis que hacen los analistas de la empresa con los datos de cada cliente, algunos de los datos que ocupan son: su consumo del anterior, su consumo de promedio, su consumo del año anterior en el mismo, las claves que aportan los lectores de la empresa, etc. Luego, existe una revisión de este análisis para algunos clientes según si el analista lo crea necesario, es decir, si sospecha que ese cliente tiene una alta probabilidad de que tenga un mal cálculo de su consumo. Luego, algunos clientes pasan nuevamente a un análisis para notar alguna irregularidad que pueda tener el cobro.

El mal cálculo de los consumos puede provocar que el cliente haga un reclamo a la empresa y puede que ese reclamo, luego de comprobabas del error, termine en una refactura. Las refacturas son básicamente el rectificar el cobro que se hace a un cliente y tienen un coste para la empresa

Estos análisis muchas veces no son suficiente para disminuir el número de refacturas, es por esto por lo que una solución prometedora es aplicar modelos de *Machine Learning* que puedan obtener patrones de clientes que muy posiblemente tengan refacturas en el futuro y así evitarlas.

Existen variados algoritmos de aprendizaje de máquina, donde varían en su eficacia o tiempos de ejecución según los datos y el problema que se les presente. Este enfoque de resolución de problema es una tendencia en Chile y el mundo. Cada día más empresas automatizan y mejoran sus sistemas para reducir costos. Bajo este contexto se hace indispensable la modernización de las distintas ramas de cada empresa para poder competir de manera eficaz con herramientas modernas.

2 Motivación y descripción del problema

Dentro de ESVAL existe un proceso definido para procesar el cobro a los clientes. Dentro de este proceso se tienen en cuenta varias variables para definir si el cobro que se está haciendo es correcto o no. Además, existe un análisis para intentar determinar los posibles errores de lectura de terreno que pueden terminan en refacturas a clientes.

2.1 Proceso de cobro a clientes y refacturas

El primer paso de este proceso es hacer las lecturas en terreno de los clientes. En este paso los lectores toman la lectura de los medidores de los clientes, además de clasificar en claves paramétricas que indican el estado del cliente en cuanto a las condiciones en terreno. Cabe destacar que estas medidas en terreno se hacen una vez al mes y cada día se toma lectura de un sector de facturación. Este sector de facturación se refiere a las medidas que se toman en un día, es decir, si se dividiera el mes en los 20 días hábiles ideales existirán 20 sectores de facturación porque los clientes a los cuales se les toma lectura dentro del mismo día en distintas localidades forman un sector.

Luego de esta primera lectura se hace un análisis de parte de los analistas de la empresa para detectar lecturas fraudulentas o erróneas con lo que, dependiendo del caso el analista puede solicitar una nueva revisión en terreno de ciertos clientes o cambiar el consumo leído en terreno comparándolo con datos de otras fuentes.

Finalmente, luego que se hacen las re-revisiones solici-

tadas se decide el cobro final a los clientes. Un diagrama simplificado de este proceso puede apreciar en la figura 1. Dentro del proceso explicado existen errores que se traducían en refacturas. Estas refacturas ocurren cuando un cliente, al tener una medida errónea o incorrecta, hace un reclamo a la empresa. La empresa toma estos reclamos y los estudia para comprobar si son verídicos o no. Luego de esta comprobación la empresa procede a hacer o no la refactura con lo que se debe devolver dinero al cliente o haciendo cobros menores en las siguientes facturas. Este problema trae consigo varios costos a la empresa en varias instancias, por ejemplo: el análisis de los reclamos o la devolución de dinero. Además de emporar la percepción general de la empresa en la comunidad.

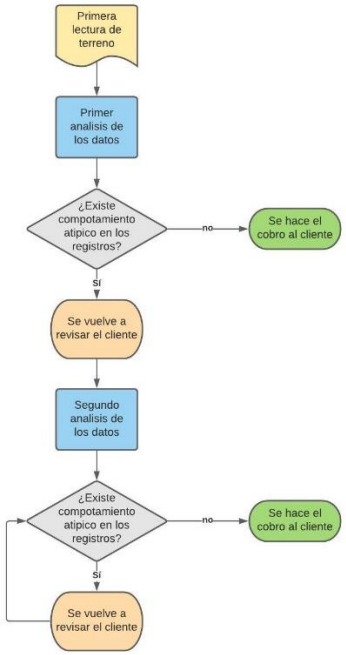

Fig. 1. Diagrama simplificado de proceso de cobro a clientes.

2.1 Solución propuesta

Para resolver o disminuir el problema de las refacturas se propone crear un modelo con algoritmos de machine learning que detecten la probabilidad de refacturas dentro del primer análisis de los datos y asistan a los analistas para tomar las decisiones pertinentes al cobro de los clientes. Esto se resumen en un problema de clasificación binario donde existirán solo dos categorías: clientes con refacturas y clientes sin refacturas.

Se espera asistir al analista en dos instancias: dentro del primer análisis y dentro del segundo análisis antes de tomar la decisión final del cobro. Un diagrama de los anterior se puede ver reflejado en la figura 2.

3 EXPLORACIÓN DE DATOS

La exploración de datos se compuso de varios pasos: determinación de tipos de datos, descripciones estadísticas, correlaciones e importancia de variables.

Fig. 2. Diagrama simplificado de proceso de cobro a clientes incluyendo los modelos de *Machine Learning*.

3.1 Descripción de los datos y valores nulos

La base de datos que se ocupará para entrenar los modelos será extraída del archivo "2016-2018v3.csv", este archivo contiene los datos de todos los clientes de las empresas ESVAL y Aguas del valle en todos los meses de los años 2016 al 2018. Este archivo contiene un total de 28 columnas con datos numéricos y categóricos, entre los que están los consumos del mes actual, consumos promedios, localidad, clave de lectura, etc. La información con todas las columnas se puede encontrar en el anexo 1 que corresponde a un informe más detallados del proyecto.

En este caso nuestra etiqueta para el entrenamiento será la columna "TIENE_REFA" que es la que describe si el cliente tuvo o no refactura el mes actual.

Dentro de los datos existen valores nulos en distintas columnas de datos categóricos que tienen que ver con las claves puestas en terreno o por los analistas de forma posterior. En conversación con los expertos de la empresa se llega a la conclusión que es conveniente llevar esos valores nulos a la categoría de que tenga mayor cantidad de registros en cada variable.

3.2 Datos atípicos

La figura 3 muestra la cantidad de refacturas de cada mes de los tres años que se incluyen en la base de datos. Se nota que existen 5 meses de distintos años que presentan un comportamiento atípico respecto a los demás.

En principio se intentó detectar cuales era las razones de estos meses atípicos y que variables indician en ellos para poder filtrarlos, y así no perder todos los datos de esos meses. Pero como se puede ver en el anexo 1, los modelos tienen mejores resultados simplemente eliminando los datos de estos meses atípicos.

3.3 Correlaciones entre variables

Para detectar correlaciones entre columnas se tomarán en cuenta solo las variables numéricas. Se hizo una matriz de correlaciones, ocupando la correlación de Pearson [1], donde las correlaciones lineales más altas se muestran de color amarillo mientras que las más bajas de color azul. Dicha matriz se puede apreciar en la figura 4.

Analizando la matriz de correlaciones se puede notar que existe poca correlación entre las variables a excepción de algunas puntuales. Estas excepciones son " LECTURA_TERRENO ", que corresponde a la lectura del mes actual con "LECTURA_ANT" y "LECTURA", además de "CONS_TERRENO_MES_ANT", que corresponde a la

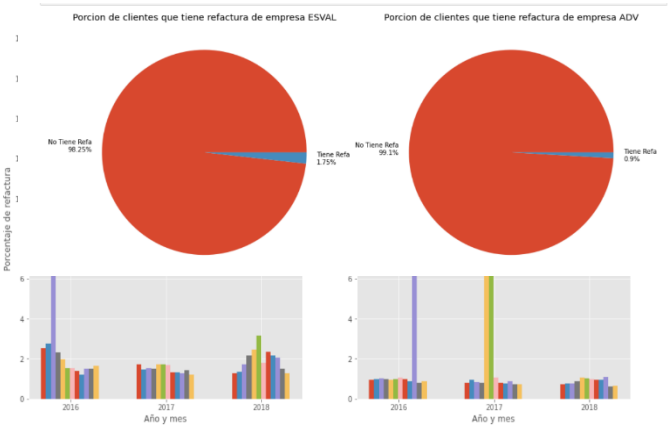

Fig. 3. Graficas de cantidad de refactura por mes y año.

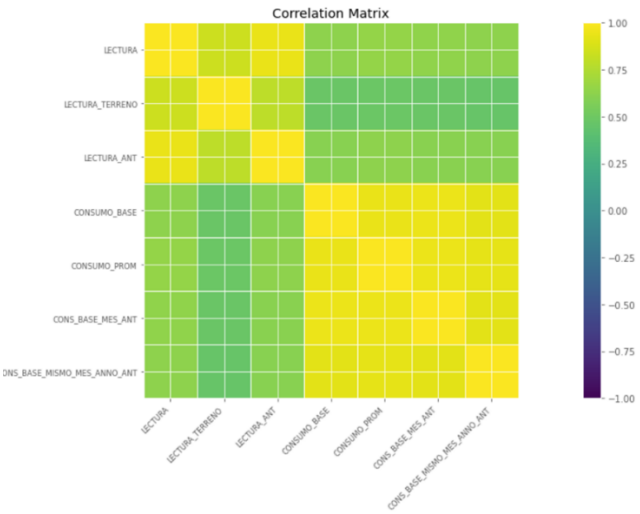

Fig. 4. Graficas de cantidad de refactura por mes y año.

lectura terreno del mes anterior al actual con "CONS_TERRENO_MES_ANT", además existe una fuerte correlación entre "CONSUMO_BASE" y "CONSUMO_PROM".

Entendiendo los datos se puede inferir que estas correlaciones son esperables, ya que corresponden a lecturas en distintas instancias o en distinto instante de tiempo, pero de los mismos clientes. Esto hace que este análisis no tengo una mayor incidencia en el preprocesamiento de los datos.

3.4 Graficas de proporciones de refacturas

Las proporciones de refactura por empresa se pueden apreciar en la figura 5. Se puede notar que este porcentaje es bastante bajo comparado con el total de registros de los datos, esto nos asegura que trabajamos con una base de datos desbalanceada. Lo anterior hace replantearse el diseño de los algoritmos, como se verá en capítulos posteriores.

En la figura 6 se muestra el porcentaje de refacturas por mes de cada empresa por separado. Se puede notar que existe un cierto aumento en los primeros meses del año que va disminuyendo a medida que este avanza. según lo conversado con expertos de la empresa esto ya se ha notado antes y se atribuye a las segundas viviendas o residencias de vacaciones que solo son ocupadas durante la estación de verano. Además, existen algunos clientes con estacionalidad contraria, es decir, se ocupan los inmuebles solo en durante el invierno y muy poco en verano, ejemplos de estos clientes podrían ser los colegios, las universidades, etc.

Los clientes que ocupan estas viviendas estacionales muy a menudo se encuentran con facturas que sospechan tienen errores, con lo que hacen un reclamo que termina en una refactura. Esta refactura puede aplazarse hasta tres meses después de ser hecho el reclamo, es por esto por lo que hasta junio existe un número más elevado de refacturas.

Esta estacionalidad de las refacturas será tomada en cuenta para la división de los datos en el preprocesamiento de estos. Sera hará un promedio de cada estación para

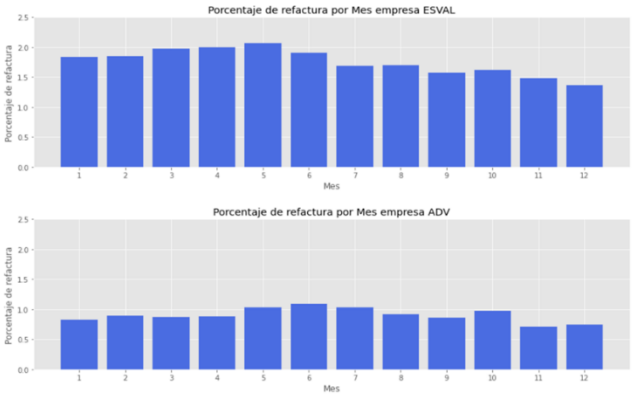

Fig. 6. Gráficos que muestran las refacturas por mes de cada empresa.

cada cliente y se calculará la diferencia entre estas. Luego, se hará una división de los clientes por estacionalidad de estos para tratar de obtener mejores resultados en los modelos que se desarrollaran posteriormente.

La figura 7 se muestra el porcentaje de refactura por tipo de cliente y por empresa. Se puede apreciar de las gráficas que en las dos empresas el tipo de cliente que más tiene refacturas son los fiscales que corresponden al

índice “F”; luego de hablar con los expertos de la empresa se notifica que a estos clientes generalmente se le hace consideraciones especiales a la hora de los cobros, y por

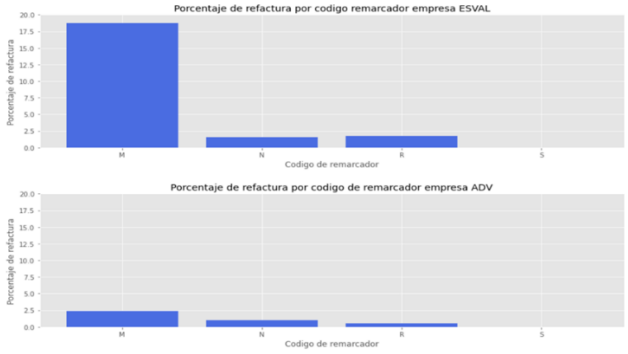

Fig. 7. Porcentaje de refacturas por código de remarcador y empresa.

eso tienen un mayor porcentaje de refacturas. Se opto por eliminar los clientes de este tipo, ya que los datos estarían sesgados gracias a estas consideraciones especiales y el porcentaje de estos clientes respecto al total es ínfimo.

3.5 análisis de componentes principales ACP

El análisis de componentes principales [2] es una proyección de las distintas variables, que conllevan distintas dimensiones, disminuyendo así las dimensiones hasta un numero donde puedan ser graficadas.

En la figura 8 se puede apreciar el análisis de componentes principales de los datos reduciéndolos a 2 dimensiones. De la gráfica se puede notar que los datos no tienen una separación línea clara, esto hace prever que

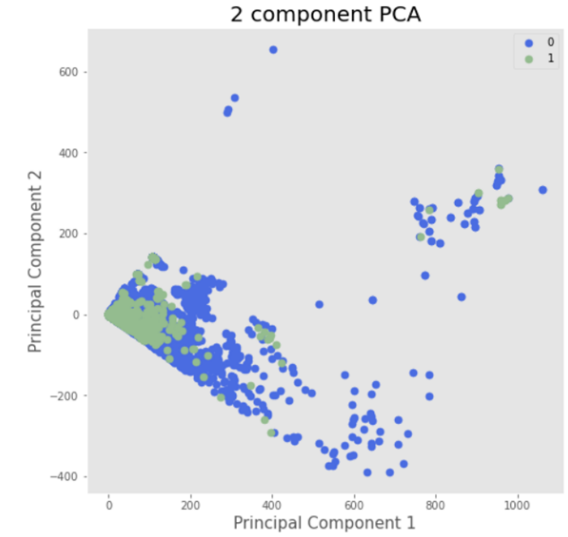

Fig. 8. Grafica de los 2 componentes principales de los datos

cuando se clasifiquen los clientes con refactura también se incurrirá en errores clasificando clientes sin refactura como si la tuvieran porque están dentro o muy cerca de los límites de los datos de clientes con refacturas.

3.5 Datos con y sin responsabilidad analista

Durante la exploración de los datos se acordó que el modelo será usado en dos instancias, como se puede ver en la figura 2. Estas instancias tienen variables que están o no disponibles. La primera instancia es en la que los datos llegan después de la primera lectura de terreno, dentro de esta no están aún las variables “LECTURA” y “CLAVE_LECTURA”. Los datos de esta instancia son llamados Datos no-analista. La segunda instancia es en la que los analistas ya han modificado o no consumos y las variables incluyen todas las columnas de los datos, pero solo los registros que tengan las categorías: "P", "A", "B", "C", "G", "M", "O", "W", y “Z” dentro de la variable “CLAVE_LECTURA”. Lo anterior, porque según los conversado con los expertos de la empresa estas claves son las que reflejan un cambio en el consumo de parte de los analistas. Estos datos son llamados datos analista.

La figura 9 muestra los porcentajes de refacturas de los datos analista y no-analista, se nota que a pesar del filtro que tienen los datos no analistas, estos igualmente tienen un numero pequeño de registros con refacturas. Los mo-

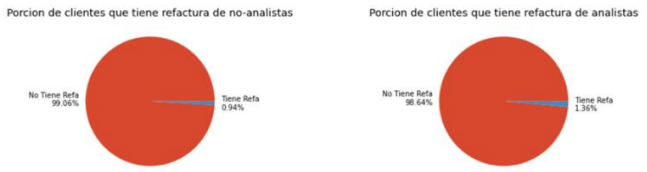

Fig. 9. Porcentaje de refacturas de datos analista y no-analistas.

delos que se diseñaran buscan mejorar lo anterior y hacer que los analistas tengan menos registro a analizar en el último paso de todo el proceso, además de asistirlos en esta decisión.

4 Preprocesamiento de datos, algoritmos y métricas

Para tener mejores resultados con los algoritmos de aprendizaje de máquina que se ocuparan, los datos fueron preprocesados mediante normalizaciones. Además, con el fin de comparar el rendimiento de los modelos se utilizaciones métricas específicas para el problema.

4.1 Codificación de datos

La codificación es una parte importante del preprocesamiento de datos dentro de esta se normalizan los datos números y se codifican los datos categóricos para que los algoritmos de aprendizaje no tengan problemas para encontrar los patrones dentro de los datos.

Los datos numéricos se codificarán con la función minimax [3], esta función normaliza los datos entre 0 y 1 dependiendo del valor máximo y mínimo de cada variable por separado .

Para las variables categóricas se hará una codificación en caliente (One Hot Encoder) [4] en la cual se agrega una columna por cada categoría de una variable (esto para todas las variables categóricas). Luego, se rellenan estas

Label Encoding

Food Name	Categorical #	Calories
Apple	1	95
Chicken	2	231
Broccoli	3	50

→ One Hot Encoding

Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

Fig. 10. Ejemplo sencillo de codificación en caliente (One Hot Encoder).

columnas con 1 o 0 dependiendo si el registro presenta esa categoría de esa variable, es decir, el registro tendrá un 0 en la columna que corresponde a una categoría que no tiene y tendrá un 1 si tiene esa categoría. La figura 10 muestra un ejemplo sencillo de esta codificación.

4.2 Eliminación de columnas

Se eliminaron algunas columnas que no tiene sentido incluir en los algoritmos porque no aportan información sobre la probabilidad que un cliente tenga una refactura. Estas columnas son:

- "NRO_SUMINISTRO": se eliminó porque es solo un identificador de cliente, pero no aporta información para la clasificación.
- "COD_LECTOR": se eliminó porque es solo un identificador de lector, pero no aporta mucha información solo la clasificación.
- "ANNO": se eliminó porque solo aporta información respecto al año de la lectura, pero en el futuro el modelo se ocupará en años que no están dentro de los datos de entrenamiento de los modelos.

También se eliminarán las columnas "CLAVE_LECTURA" y "LECTURA" en los datos de no-analista.

4.3 División de datos

En algunas ocasiones la división de los datos tiene buenos resultados a la hora de entrenar los modelos, porque estos se comportan mejor con datos más homogéneos a la hora de buscar patrones de clasificación. Es por esto por lo que se probó con varias divisiones de los datos para comprobar si realmente los modelos se comportan de mejor manera. Los mejores resultados de estas pruebas lo tuvieron la división por código de diámetros, es decir, la columna "COD_DIAMETRO" y por temporada de consumo. Para hacer la última separación dicha se necesita crear una nueva variable, proceso que se describe a continuación.

Después de probar varias divisiones de los datos y tener una conversación con los expertos de la empresa se llega a la conclusión que vale la pena probar una división por localidad y creando una nueva columna de las temporadas. Esta nueva columna se realizó sumando de cada cliente todos los consumos de los meses de la temporada de verano (diciembre, enero, febrero y marzo) y dividiéndolo por el total de ellos. Así, se hizo lo mismo para la temporada de invierno-otoño-primavera. Luego, se hizo una diferencia entre los consumos promedio de las dos temporadas para crear una nueva columna "DIFF_CONSUMOS". La fórmula para explicar el procedimiento anterior seria:

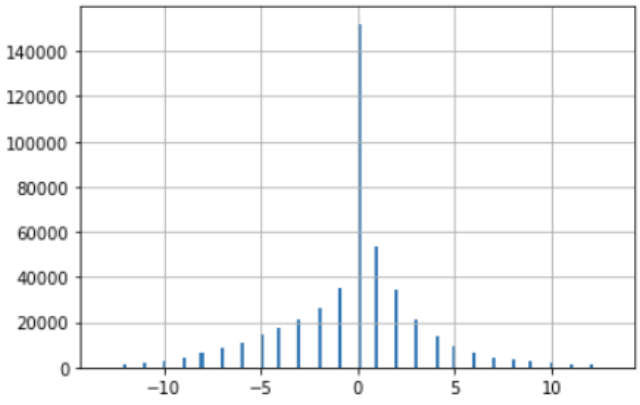

Fig. 11. Histograma con nueva columna "DIFF_CONSUMO".

$$diff_{consumo} = \left(\frac{Dic + Ene + Feb + Mar}{4}\right) - \left(\frac{Abr + May + Jun + Jul + Ago + Sep + Oct + Nov}{8}\right)$$

Donde Dic, Ene. Feb, Mar, Abr, May, Jun, Jul, Ago, Sep, Oct y Nov corresponden a los consumos promedios de sus meses respectivos de todos los años. La figura 11 muestra un histograma con esta nueva columna.

4.4 Algoritmos de machine learning empleados

Se probaron cuatro algoritmos de machine learning para encontrar una solución al problema de este trabajo que son descritos también en el anexo 1. Se utilizo la **regresión logística [5]** es un algoritmo de machine learning que intentar separa los planos de n dimensiones de manera que los registros de cada etiqueta estén separados por un umbral de decisión. Este algoritmo funciona de forma iterativa: primero calculando el error de clasificación de los datos dependiendo de una función de costo, luego de pasar por una función de decisión que en general es la función logística (o sigmoide). Con el error calculado intenta ajustar los parámetros con el gradiente descendiente para así mejorar la eficacia de la clasificación. En general, se le da un número máximo de iteraciones para no caer en sobreajuste de los modelos.

También se utilizó el algoritmo de **árbol de decisión [6]** para clasificación es un algoritmo que dependiendo de una condición en un nodo va decidiendo como se clasificando los datos o como se van repartiendo a lo largo del árbol hasta llegar a los nodos o hojas finales. Cabe destacar que dependiendo de cómo se ordenen las condiciones y las variables en un árbol puede tener resultados distintos. Este algoritmo va probando con cierta cantidad de

datos en cada nodo cual sería la mejor separación y porque variables hacerlo.

Además, se utilizaron las **redes neuronales artificiales [7] que** son un algoritmo que intenta imitar el comportamiento de las neuronas de nuestro cerebro. Este algoritmo es iterativo: donde luego de definirse cuantas capas y neuronas tendrá se hace pasar todos los datos por el modelo para luego propagar el error hacia atrás e ir cambiando los parámetros de cada neurona (que serían los pesos de cada variable). El pasar todos los datos por el modelo se le llama época, en general el modelo se entrena con varias épocas para encontrar el valor más cerca del mínimo global. Esta adecuación de los parámetros se hace con la función del gradiente descendiente y con una función de costo.

Y, por último, se utilizó el algoritmo **XGBoost [8]** que es un algoritmo de booster, esto quiere decir que es un algoritmo que tomar varios algoritmos simples y toma una decisión dependiendo lo que diga la mayoría de estos algoritmos por separado. En este caso, XGBoost crea varios árboles de decisión débiles para clasificar los datos.

4.5 Métricas de evaluación

Todas las métricas nacen de la matriz de confusión [9]. Esta matriz tiene como elementos las clasificaciones del algoritmo, estos elementos son:

- Verdaderos positivos (TP): son la cantidad de datos de la clase positiva (en este caso, los clientes con refacturas) que son clasificados correctamente.
- Falsos positivos (FP): son la cantidad de datos de la clase positiva que son clasificados incorrectamente.
- Verdaderos negativos (TN): son la cantidad de datos de la clase negativa (en este caso, los clientes sin refactura) que son clasificados correctamente.
- Falsos negativos (FN): son la cantidad de datos de la clase negativa que son clasificados incorrectamente.

La figura 12 muestra una matriz de confusión para el caso de clasificación binaria, como es nuestro caso.

Se define dos métricas generales que servirán para definir las métricas específicas de nuestro problema. El recall [9] y la precisión [9] que se ocupan más normalmente en modelos con bases de datos desbalanceados. El primero se define con la siguiente formula:

$$recall = \frac{VP}{VP + FN}$$

Mientras que la precisión lo hace con:

$$precision = \frac{VP}{VP + FP}$$

Estas dos métricas tienen cierta relación, cuando una aumenta la otra tiende a disminuir. Esto es porque las dos toman en cuenta los VP, pero dependiendo de parámetros

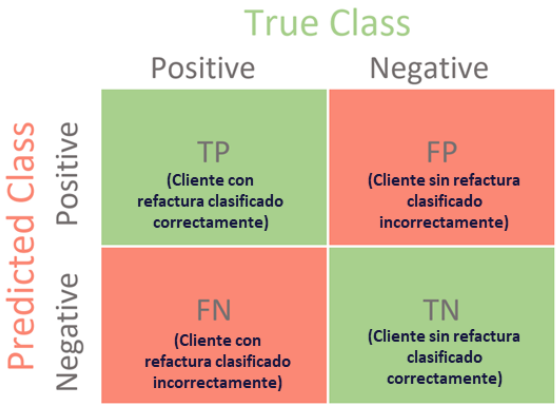

Fig. 12. Matriz de confusión para clasificación binaria.

distintos. En el caso del recall este es alto cuando se detecta bien la clase, es decir cuando los FN no son muchos. Pero, no toma en cuenta los FP. Mientras que en el caso de la precisión esta es alta cuando se clasifica la clase positiva con más precisión, es decir cuando los FP no son mucho. Pero, no toma en cuenta los FN. Como no se puede tener una visión clara de cual modelo es mejor solo con estas métricas, se agregan más.

F1-score [9] toma un promedio entre recall y precisión para dar una idea aproximada del rendimiento de los modelos. Se define con la siguiente formula:

$$F1 - score = \frac{2 * (precision * recall)}{precision + recall}$$

Por otro lado, la métrica G-mean [9] también toma un promedio, pero de la clasificación de cada clase para tener una idea aproximada de que tan bien un modelo clasifica cada clase por separado. Esta métrica se define con la siguiente formula:

$$G - mean = \sqrt{\frac{VP}{VP + FN} + \frac{VN}{VN + FP}}$$

Estas métricas, a pesar de dar más información que el recall y la precisión, tiene un problema: ¿realmente importa el recall y la precisión de la misma manera? ¿realmente importa la clasificación de ambas clases de la misma manera? La respuesta a estas preguntas es no, porque lo que más nos interesa es clasificar de buena manera los datos con la etiqueta de clientes con refactura.

Para responder a las falencias de las métricas anterior se investigó una nueva métrica que puede sincronizarse con nuestros requerimientos respecto a la importancia de cada variable en nuestra clasificación. Esta nueva métrica es F-measure [9] que se define a continuación:

$$F_\beta = \frac{(1 + \beta)^2 * (recall * precision)}{\beta^2 * precision + recall}$$

Si el valor de β es mayor que 1 se le da más importancia al recall (es decir, no importa que haya mucho positivos mal clasificados mientras no haya muchos negativos mal clasificados), mientras que si β es menor que 1 se da más importancia a la precision (es decir, no importa que haya varios negativos mal clasificados, mientras que los positivos estén bien clasificados).

Otra métrica para mejorar las falencias de las métricas anteriores es AUC (Area Under Curve) [9], esta depende de la curva ROC [9]. La curva ROC es un gráfico que muestra el rendimiento de un modelo de clasificación en todos los umbrales de clasificación. Mientras que la métrica AUC es el área bajo la curva ROC, donde un área de 1 es un clasificador perfecto y 0 uno que clasifica todo mal.

La anterior curva ROC descrita depende de la tasa de verdaderos positivos (TPR) y la tasa de falsos positivos (FPR) que son definidos a continuación:

$$TPR = \frac{VP}{VP + FP}$$

$$FPR = \frac{FP}{FP + VN}$$

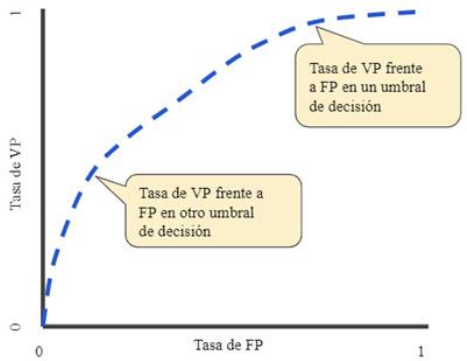

Fig. 13. Ejemplo de curva ROC.

La figura 13 muestra un ejemplo de la curva ROC,

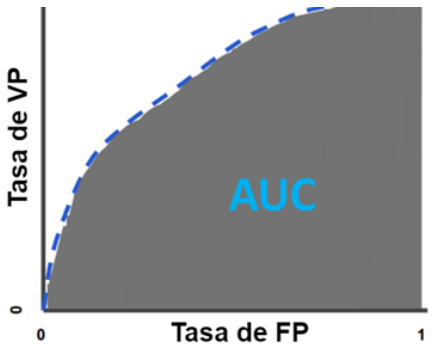

Fig. 14. Ejemplo de métrica AUC.

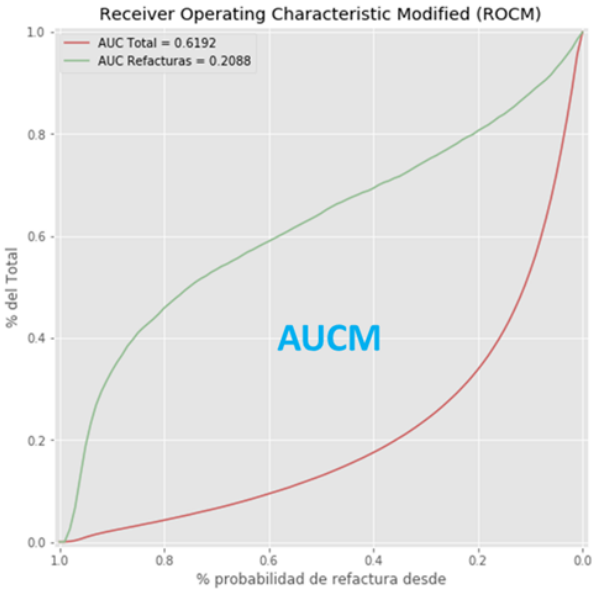

Fig. 15. Ejemplo de métrica AUC.

mientras que la figura 14 muestra un ejemplo del área que sería nuestra métrica AUC.

Además, se creó una nueva métrica que toma en cuenta las características de la empresa. La métrica AUCM (Area Under Curve Modified) es una métrica inventada para el propósito de este trabajo. La figura 15 muestra un ejemplo de estas métricas. Donde la línea roja (AUC total) se refiere a la cantidad de registros que se revisan con todos los umbrales de decisión, mientras que la línea verde (AUC refacturas) se refiere al porcentaje de refacturas (verdaderos positivos) que son revisadas con cada umbral de decisión.

Cabe destacar que después de hablar con los expertos de la empresa se llegó a la conclusión que más que mostrar el total de los umbrales de decisión, realmente importa cuantas refacturas son revisadas con el total de registros que pueden revisar los analistas. Así con 5 analistas, se pueden revisar un total de 3500 registros.

5 Modelos, resultados y tiempos

En la figura 2 se puede notar que el modelo tendrá dos instancias de uso y se entrenaron modelos para cada instancia. En la primera instancia se ocupan solos los datos de la lectura de terreno y son llamados datos no-analista. Mientras que en la segunda instancia se ocupan los datos de la lectura de terreno y los que aportan los analistas, estos datos son llamados datos analista. Además, se presenta para cada instancia modelos con datos sin dividir, con división por código de diámetro y con división por temporada de consumo.

Según lo hablado con los expertos de la empresa una división que tiene mucho sentido es la por temporada.

Tabla 1: Resultados de modelos sin divisiones del mes de diciembre del año 2018.

sector	XGBoost sin división		Neural Networks sin división	
	Umbral decisión	% refacturas encontradas	Umbral decisión	% refacturas encontradas
1	0,39	0,50	0,59	**0,76**
2	0,49	**0,57**	0,75	0,53
3	0,38	**0,49**	0,67	0,46
4	0,64	0,33	0,65	**0,44**
5	0,58	0,38	0,69	**0,41**
6	0,68	0,49	0,68	**0,62**
7	0,74	0,45	0,69	**0,59**
8	0,65	0,34	0,57	**0,41**
9	0,49	**0,60**	0,7	0,50
10	0,42	0,41	0,67	**0,41**
11	0,66	0,34	0,72	**0,49**
12	0,49	0,32	0,64	**0,38**
13	0,51	**0,36**	0,65	0,36
14	0,34	0,69	0,56	**0,70**
15	0,56	**0,56**	0,65	0,51
16	0,48	0,22	0,62	**0,37**
17	0,42	**0,47**	0,66	0,45
18	0,44	0,64	0,58	**0,64**
19	0,51	0,61	0,72	**0,68**
20	0,28	0,41	0,66	**0,48**
Promedio		0,46		**0,51**
Máximo		0,69		**0,76**
Mínimo		0,22		**0,36**

Esto se nombra en el subcapítulo 3.1.3, donde se definió una nueva columna "DIFF_CONSUMOS" y en la figura 11 se presenta un histograma de esta nueva columna. Se nota que existe tres grupos marcados:

- Los clientes que tienen más consumos en la temporada de verano.
- Los clientes que tienen más consumos en la temporada del resto de años.
- Los clientes que tienen el mismo consumo en ambas temporadas.

Como se dijo en el capítulo 4, estos tres grupos son esperables, ya que existen clientes que solo ocupan los inmuebles en verano (inmuebles de veraneo), otro que la ocupan solo en el resto del año (colegios, universidades) y otros que tienen un consumo constante a lo largo del año (casas residenciales u otros clientes.

Dado todo lo anterior se hicieron tres modelos distintos para cada tipo de clientes según consumo por temporada y se evaluaron los resultados en los datos de diciembre del año 2018.

5.1 Modelos de datos no-analista

Estos modelos ocupan los datos no-analista y serán usados en la primera instancia de nuestro sistema. En la tabla 1 se pueden apreciar los resultados de estos modelos sin división, mientras que en la tabla 2 se pueden apreciar

Tabla 2: Resultados de modelos con división por temporada del mes de diciembre del año 2018.

sector	XGBoost por temporada		Neural Networks por temporada	
	Umbral decisión	% refacturas encontradas	Umbral decisión	% refacturas encontradas
1	0,52	0,54	0,61	**0,73**
2	0,6	0,59	0,72	**0,60**
3	0,54	0,41	0,62	**0,53**
4	0,75	0,33	0,61	**0,45**
5	0,65	0,45	0,67	**0,48**
6	0,75	0,56	0,71	**0,61**
7	0,78	0,41	0,7	**0,45**
8	0,76	0,32	0,6	**0,43**
9	0,61	0,52	0,73	**0,70**
10	0,69	0,38	0,59	**0,46**
11	0,74	0,38	0,76	**0,43**
12	0,71	0,33	0,65	**0,51**
13	0,62	0,34	0,59	**0,38**
14	0,45	0,66	0,56	**0,75**
15	0,69	0,57	0,71	**0,57**
16	0,64	0,34	0,7	**0,48**
17	0,59	0,43	0,71	**0,48**
18	0,58	0,63	0,61	**0,65**
19	0,64	0,55	0,66	**0,74**
20	0,54	0,41	0,63	**0,49**
Promedio		0,463		**0,55**
Máximo		0,66		**0,75**
Mínimo		0,32		**0,38**

los resultados de los modelos con división. Comparando las dos tablas se puede ver que los modelos con divisiones por temporada tienen un mejor desempeño en la mayoría de las localidades, a excepción de algunos casos especiales

Además, de las tablas nombradas se puede notar que existe una gran variabilidad del desempeño de los modelos con respecto al sector de facturación, según los expertos de la empresa esto puede deberse a los distintos que pueden ser algunos sectores dentro de cada localidad. Lo que se busca finalmente con los modelos es tratar de tener el mejor desempeño en todos los sectores por separado. Asumiendo esto, es una buena opción ocupar distintos modelos a la hora de clasificar datos nuevos y ocupar el modelo que mejor se acomode a cada sector de facturación.

Del análisis hecho del problema por parte del investi-

gador se notó que una variable apropiada para segmentar los clientes es el código de diámetro, ya que entrega información del remarcador (diámetros medianos implican clientes con matrices) y de tipo de cliente (diámetros pequeños implican clientes residenciales y muy grandes implican clientes industriales). Para separar los datos por código de diámetro se utilizó un umbral. Como los buenos resultados de esta división fue insinuada por lo análisis, se probó varios umbrales para segmentar a los clientes y se llegó a que el óptimo era el de "50". Este umbral separa los datos en dos grupos: El primer grupo con código de diámetro menor a "50" y el segundo grupo con código de diámetros mayor a "50".

La tabla 3 muestra los resultados por sector de facturación de los modelos divididos por código de diámetros de diciembre del año 2018. Se nota que dentro de cada sector de facturación existen clientes dentro de los dos grupos, asique para hacer las pruebas se dividieron los datos dentro de cada sector de facturación, se clasifico con el modelo y luego se volvieron a unir los datos.

Tabla 3: Resultados de modelos con división por código de diámetro del mes de diciembre de 2018.

sector	XGBoost por diámetro		Neural Networks por diámetro	
	Umbral decisión	% refacturas encontradas	Umbral decisión	% refacturas encontradas
1	0,49	0,52	0,6	**0,79**
2	0,59	**0,71**	0,72	0,61
3	0,44	**0,58**	0,6	0,51
4	0,75	0,34	0,6	**0,47**
5	0,53	**0,51**	0,7	0,50
6	0,72	0,59	0,72	**0,65**
7	0,81	0,39	0,68	**0,48**
8	0,76	0,39	0,6	**0,48**
9	0,7	**0,80**	0,66	0,56
10	0,55	0,45	0,65	**0,46**
11	0,77	0,31	0,72	**0,47**
12	0,39	0,40	0,65	**0,47**
13	0,59	0,34	0,62	**0,39**
14	0,35	0,66	0,54	**0,79**
15	0,82	0,42	0,74	**0,6**
16	0,78	0,27	0,71	**0,39**
17	0,77	0,38	0,71	**0,55**
18	0,7	0,63	0,64	**0,66**
19	0,44	0,63	0,59	**0,77**
20	0,37	0,42	0,6	**0,49**
Promedio		0,49		**0,55**
Máximo		**0,80**		0,79
Mínimo		0,27		**0,39**

La tabla 4 muestra las métricas promedio de los datos sin divisiones, así como también los con divisiones por temporada y por código de diámetro. De esta tabla se puede notar que en casi todas las métricas el modelo de redes neuronales con los datos divididos por temporada tiene los mejores resultados. Aunque el modelo de redes neuronales con los datos divididos por diámetros tiene un mejor resultado la métrica de porcentaje de refacturas revisadas. La métrica anterior es la más importante en términos de efectividad del modelo, es por esto por lo que no podría decirse cuál de los dos modelos es mejor. Con lo anterior presente se propone tener dos modelos para la clasificación, así el analista puede contrastar los resultados de estos dos modelos y tomar una decisión frente al cobro a los clientes.

Tabla 4: Comparación de promedios de resultados de modelos sin división, con división por temporada y con división por código de diámetro.

Modelos	F1-score	G-mean	F-measure	AUCM	Porcentaje de refacturas revisadas
NN sin división	0,10	0,75	0,68	0,38	0,53
XGB sin división	**0,11**	0,66	0,54	0,35	0,48
NN por temporada	0,1	**0,76**	**0,69**	**0,40**	0,55
XGB por temporada	0,08	0,69	0,58	0,37	0,46
NN por diámetro	0,1	0,75	0,69	0,41	**0,55**
XGB por diámetro	0,101	0,71	0,63	0,39	0,49

5.2 Modelos de datos analista

Estos modelos serán ocupados en la segunda instancia del sistema con los datos analista. Los resultados de los modelos con estos datos son bastante similares a los resultados con los datos no-analista, es por esto por lo que se enfocara en los modelos que dieron resultados. Cabe destacar que se hicieron las mismas pruebas con estos datos que con los datos no-analista.

En estos modelos no tiene sentido ocupar la métrica de porcentaje de refacturas revisadas porque serán ocupados ya con los datos analizados y filtrados, esto por esto que las comparaciones entre estos modelos se hacen con las métricas más importantes, en este caso F-Measure y AUCM; también, por sector de facturación.

En la tabla 5 se muestran los resultados de los modelos sin división por sector de facturación. Se nota de la tabla que el modelo con mejores resultados es el del algoritmo de redes neuronales, pero solo en algunos sectores.

La tabla 6 muestra los resultados de los modelos con división por temporada de los datos analista. Como se dijo anteriormente se evaluarán los modelos con las métricas más importantes. Se nota de la tabla que los modelos tienen resultados muy parecidos entre sí, ya que dependiendo del sector uno u otro algoritmo puede ser óptimo. En este caso, quizá la mejor forma de comparar-

los será con el promedio de las métricas. Esto se hare posteriormente para tener una idea del mejor modelo comparando todos ellos de una vez.

Tabla 5: Resultados de modelos con datos analista sin dividir del mes de diciembre del año 2018.

sector	XGboost sin división		Neural Networks sin división	
	F-Measure	AUCM	F-Measure	AUCM
1	0,66	0,35	**0,77**	**0,45**
2	**0,68**	0,36	0,60	**0,38**
3	0,49	0,30	**0,50**	**0,32**
4	0,53	0,27	**0,77**	**0,35**
5	0,50	0,23	**0,97**	**0,47**
6	0,63	0,35	**0,71**	**0,38**
7	**0,67**	0,30	0,65	**0,36**
8	0,66	0,38	**0,74**	**0,43**
9	0,99	**0,53**	**0,89**	0,52
10	**0,60**	**0,39**	0,52	0,36
11	0,72	0,39	**1,01**	**0,52**
12	**0,79**	**0,45**	0,67	0,37
13	0,46	0,19	**0,58**	**0,32**
14	0,59	0,38	**0,65**	**0,40**
15	0,70	0,37	**0,73**	**0,38**
16	0,45	**0,30**	0,70	0,29
17	0,47	0,23	**0,51**	**0,30**
18	0,55	0,34	**0,67**	**0,42**
19	**0,69**	0,34	0,67	**0,36**
20	**0,56**	**0,30**	0,50	0,29
Promedio	0,62	0,34	**0,69**	**0,38**
Máximo	0,99	**0,53**	**1,01**	0,52
Mínimo	0,45	0,19	**0,50**	**0,29**

La tabla 7 muestra los resultados de los modelos de datos analista con división por diámetro. De esta tabla se puede notar que el mejor modelo lo tiene el con redes neuronales, ya que tiene un promedio mayor en las métricas lo que refleja un mejor comportamiento en la mayoría de los sectores de facturación.

Finalmente, en la tabla 8 se muestran los promedios de todas las métricas de todos los modelos mencionados con los datos analista, de esta tabla se puede notar que el mejor modelo es el del algoritmo de redes neuronales con la división por diámetro, obteniendo mejores resultados en todas las métricas.

Pero, cabe destacar que estos resultados no reflejan la eficacia de los modelos específicamente en cada sector de facturación, por esto una propuesta seria que se ocuparan variados modelos dependiendo de a qué sector de facturación corresponda el cliente a analizar. Otra opción sería tomar los dos mejores modelos y dejarlos para el uso diario del sistema, así el analista tendría dos "opiniones" distintas para poder tomar una decisión más acertada sobre la refactura de los clientes.

Tabla 6: Resultados de modelos con datos analista con división por temporada del mes de diciembre del año 2018.

sector	XGBoost por temporada		Neural Networks por temporada	
	F-Measure	AUCM	F-Measure	AUCM
1	0,74	0,36	**0,754**	**0,48**
2	**0,63**	0,38	0,62	**0,40**
3	**0,51**	0,23	0,50	**0,33**
4	0,64	0,27	**0,74**	**0,35**
5	0,59	0,24	**0,78**	**0,39**
6	**0,69**	0,34	0,66	**0,37**
7	**0,70**	**0,32**	0,59	0,32
8	**0,74**	0,32	0,61	**0,38**
9	0,86	0,42	**0,97**	**0,59**
10	**0,60**	**0,38**	0,45	0,28
11	**0,88**	0,40	0,74	**0,44**
12	**0,74**	0,36	0,58	**0,39**
13	**0,62**	0,26	0,59	**0,37**
14	**0,78**	**0,460**	0,73	0,46
15	0,71	0,378	**0,71**	**0,43**
16	**0,65**	0,33	0,61	**0,34**
17	**0,55**	0,21	0,54	**0,26**
18	0,61	0,298	**0,69**	**0,44**
19	**0,80**	**0,348**	0,60	0,27
20	**0,66**	**0,36**	0,55	0,35
Promedio	**0,69**	0,33	0,65	**0,38**
Máximo	0,88	0,46	**0,97**	0,59
Mínimo	**0,51**	0,21	0,45	0,26

5.3 Tiempos de ejecución

Los tiempos de ejecución en cuanto a entrenamientos y prueba de los modelos son importantes, ya que la idea de este proyecto es poder reentrenarlo cada cierto tiempo para mejorar su desempeño y tener datos, y por lo tanto patrones, más actualizados. En principio lo ideal sería poder reentrenar los modelos en un fin de semana, es decir, aproximadamente en 60 horas.

Los tiempos de prueba son los tiempos promedio que demora el modelo en clasificar los datos de un sector de facturación del mes de diciembre del año 2018.

Se ejecutaron los programas de entrenamiento y prueba en una computadora con procesador AMD Ryzen 5 4600H de 3.00Hz con 6 núcleos y 12 hilos, 16 Gb de memoria y una tarjeta de video NVIDIA GeForce GTX 1650 con 4 Gb de memoria.

En la tabla 4-12 se muestran los tiempos de entrenamiento de los modelos con datos no-analista, mientras que la tabla 4-13 muestra los tiempos de entrenamiento de los modelos con datos analista. Es clara la diferencia de tiempos entre las dos tablas, esto porque los datos

analista son considerablemente menos que los no-analista.

Tabla 7: Resultados de modelos con datos analista con división por diámetro datos de diciembre del 2018.

sector	XGBoost con división		Neural Networks con división	
	F-Measure	AUCM	F-Measure	AUCM
1	0,80	**0,43**	**0,81**	0,42
2	0,65	0,42	**0,76**	**0,42**
3	0,62	0,38	**0,67**	**0,44**
4	0,57	0,27	**0,67**	**0,36**
5	0,65	0,28	**0,97**	**0,53**
6	0,76	0,38	**0,86**	**0,50**
7	0,61	0,34	**0,65**	**0,35**
8	0,77	0,45	**0,85**	**0,55**
9	0,93	0,46	**0,99**	**0,55**
10	**0,55**	**0,38**	0,52	0,35
11	0,77	0,35	**0,82**	**0,48**
12	**0,82**	**0,47**	0,69	0,41
13	0,55	0,31	**0,69**	**0,47**
14	0,62	0,36	**0,82**	**0,51**
15	0,70	0,41	**0,77**	**0,49**
16	0,64	0,37	**0,67**	**0,38**
17	0,59	0,27	**0,69**	**0,35**
18	0,61	0,38	**0,82**	**0,55**
19	**0,80**	0,40	0,76	**0,44**
20	0,67	0,36	**0,84**	**0,43**
Promedio	0,68	0,37	**0,77**	**0,45**
Máximo	0,93	0,47	**0,99**	**0,55**
Mínimo	**0,55**	0,276	0,52	**0,35**

Tabla 8: Comparación de promedios de resultados de modelos sin división, con división por temporada y con división por código de diámetro.

Modelos	F1-score	G-mean	F-measure	AUCM
NN sin división	0,08	0,74	0,69	0,38
XGB sin división	0,08	0,71	0,62	0,34
NN por temporada	0,09	0,72	0,65	0,38
XGB por temporada	0,06	0,74	0,69	0,33
NN por diámetro	**0,10**	**0,79**	**0,77**	**0,45**
XGB por diámetro	0,07	0,74	0,68	0,37

De las tablas anteriores se puede notar que el modelo con redes neuronales tiene un tiempo de entrenamiento mucho mayor que el del algoritmo *XGboost*. En principio, esto no sería un problema, pero si los tiempos de entrenamiento combinados de los modelos sobrepasan las 60 horas máximas mencionadas para entrenamiento habría que analizar eliminar un modelo. Observando los tiempos de entrenamiento se nota que sin problema podrían entrenarse dos modelos de datos analista y dos modelos de data no-analista.

Por último, cabe destacar que los tiempos de entrenamiento y de ejecución del día a día podrían tener variaciones dependiendo del equipo que se esté utilizando para correrlos.

Tabla 9: Comparación de tiempos de entrenamiento y prueba de los modelos con datos no-analista.

Modelos	Tiempo entrenamiento (hh:mm:ss)	Tiempo prueba (hh:mm:ss)
NN sin división	12:03:54	00:01:46
XGB sin división	01:15:53	00:00:35
NN por temporada	15:44:36	00:00:20
XGB por temporada	01:57:19	00:00:15
NN por diámetro	37:04:13	00:00:08
XGB por diámetro	02:02:32	00:00:17

Tabla 10: Comparación de tiempos de entrenamiento y prueba de los modelos con datos analista.

Modelos	Tiempo entrenamiento (hh:mm:ss)	Tiempo prueba (hh:mm:ss)
NN sin división	00:26:59	00:00:02
XGB sin división	00:04:29	00:00:09
NN por temporada	00:33:47	00:00:35
XGB por temporada	00:01:05	00:00:04
NN por diámetro	00:34:23	00:00:02
XGB por diámetro	00:06:27	00:00:01

5 CONCLUSIONES

En este trabajo se hizo un recorrido por las distintas aristas del problema de refacturas, dejando claro los costos que tiene para la empresa. Además, se hizo una exploración detallada de los datos para obtener funciones que automaticen la limpieza de esto y los codifiquen para el buen funcionamiento de los modelos de *Machine Learning* diseñados. También, se separaron los datos por datos analista y datos no-analista, según lo conversados con expertos de la empresa.

Se crearon modelos de *Machine Learning* para solucionar el problema mencionado. Se hicieron pruebas con modelos de árboles de decisión, regresión logística, redes neuronales y *XGBoost*. Se notó que los mejores modelos se obtenían con los dos ultimo algoritmo mencionados.

Se probó con varias divisiones de los datos para aislar posibles patrones que pudieron ser encontrados por los algoritmos de aprendizaje. Finalmente, después de varias pruebas se notó que las mejores divisiones de los datos son por temporada y por código de diámetro.

Además, se estudiaron los tiempos de ejecución del entrenamiento y prueba de los modelos, llegando a la

conclusión que se podrían entrenar perfectamente dos modelos de datos analista y dos modelos de datos no-analista.

Según los analizado en este trabajo y los conversado con los expertos de la empresa se proponen distintas mejoras o modificaciones al proyecto. Primeramente, el proyecto se implementará en la nube para un funcionamiento más eficaz. Se nota que a pesar de que los tiempos de entrenamientos de los modelos son elevados y necesitan de una implementación en la nube, los tiempos de prueba son bastante bajos por lo que se hace posible implementarlos en computadoras personales. Como propuesta más futura, se espera poder idear una implementación en teléfonos celulares para que así los analistas tengan aún más a mano la asistencia de los modelos.

Además, se propone que se ocupen varios modelos en la ejecución del sistema día a día, pero que solo arroje un resultado. Con esto se pretende acotar la complejidad de la decisión del analista y pueda apoyar de mejor manera en el sistema de aprendizaje de máquina. Esto se haría decidiendo que modelos es mejor para un sector de facturación desdiciendo en la fase de entrenamiento, así el objetivo final sería implementar esta optimización de modelo por sector de manera automática cada vez que se reentrenen los modelos.

REFERENCIAS

[1] Benesty, J., Chen, J., Huang, Y., & Cohen, I. (2009). Pearson correlation coefficient. In Noise reduction in speech processing (pp. 1-4). Springer, Berlin, Heidelberg.

[2] Maćkiewicz, A., & Ratajczak, W. (1993). Principal components analysis (PCA). Computers & Geosciences, 19(3), 303-342.

[3] Farnia, F., & Tse, D. (2016). A minimax approach to supervised learning. In Advances in Neural Information Processing Systems (pp. 4240-4248).

[4] Rodríguez, P., Bautista, M. A., Gonzalez, J., & Escalera, S. (2018). Beyond one-hot encoding: Lower dimensional target embedding. Image and Vision Computing, 75, 21-31.

[5] ovell, A. J. (1995). Análisis de regresión logística. Madrid: Centro de Investigaciones Sociológicas.

[6] Safavian, S. R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. IEEE transactions on systems, man, and cybernetics, 21(3), 660-674.

[7] Wang, S. C. (2003). Artificial neural network. In Interdisciplinary computing in java programming (pp. 81-100). Springer, Boston, MA.

[8] Zhang, L., & Zhan, C. (2017, May). Machine learning in rock facies classification: an application of XGBoost. In International Geophysical Conference, Qingdao, China, 17-20 April 2017 (pp. 1371-1374). Society of Exploration Geophysicists and Chinese Petroleum Society.

[9] Branco, P., Torgo, L., & Ribeiro, R. (2015). A survey of predictive modelling under imbalanced distributions. arXiv preprint arXiv:1505.01658.

Joaquín Farias Muñoz Estudiante de ingeniería civil electrónica, Pontificia Universidad católica de Valparaíso. Practicante ESVAL S.A.