Joaquín Esteban Farias Muñoz

Modelos de machine learning aplicado al cálculo de probabilidad de refacturas de clientes

Informe Proyecto Machine Learning ESVAL

Valparaíso, 03 de enero de 2021

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 claves: *Machine Learning*, refacturas, ESVAL, aprendizaje supervisado, clasificación.

Índice general

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 calculo 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 su eficacia y/o tiempos de ejecución varían 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.

Objetivos generales

- Crear modelos de machine learning para detectar el error de lectura en terreno y en las modificaciones aplicadas por el analista de consumos a partir de las refacturas aplicadas a posteriori.
- Analizar mejores modelos y tiempos de ejecución de estos.

Objetivos específicos

- Explorar los datos de entrenamiento de los modelos.
- Filtrar, limpiar y codificar los datos. Además, encontrar posibles nuevas variables que podrías ser de utilidad.
- Crear modelos con algoritmos de árboles de decisión, regresión logística, redes neuronales artificiales y XGboost para la clasificación de clientes con refactura.

1 Definición del problema

Este capítulo se abordarán las características del problema que se intenta solucionar explicando las complejidades de este y el proceso donde estaría inmersa la solución .

1.1 Proceso de cobro y refacturas

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.

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 mediciones en terreno se hacen una vez al mes y cada día se toma lectura de un sector de facturación distinto. 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 solicitadas se decide el cobro final a los clientes. Un diagrama simplificado de este proceso puede apreciar en la *Figura 1-1*.

Cabe destacar que dentro del resumen anterior del proceso se omitieron varios pasos intermedios respecto al análisis de los datos. Además, también es bueno aclarar que este proceso conlleva errores que se traducen en refacturas, estas tienen un costo para la empresa, una baja en la satisfacción de los clientes y un efecto negativo en la percepción general de la empresa.

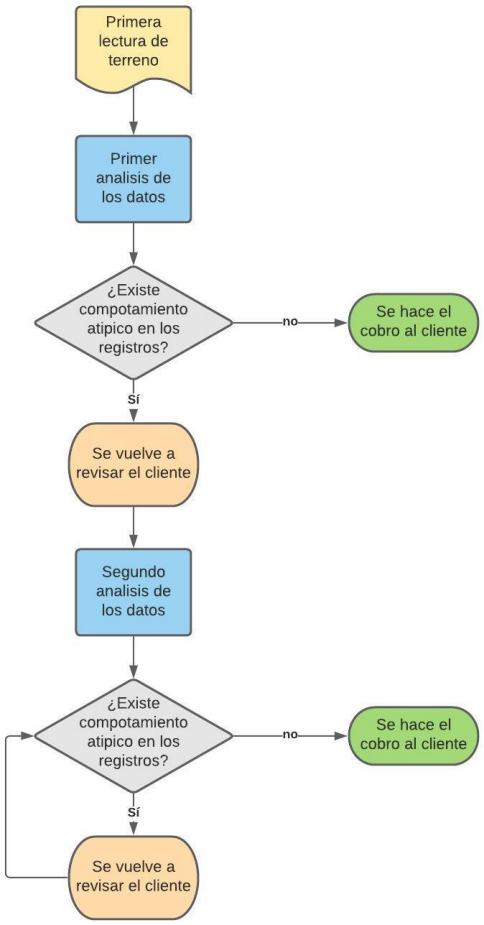

Figura 1-1: Diagrama de flujo simplificado de proceso de cobro a clientes.

1.2 Problema: Refacturas

Dentro del proceso explicado en el subcapítulo anterior se dijo que existían 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 empeorar la percepción general de la empresa en la comunidad.

1.3 Soluciones propuestas

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 1-2*.

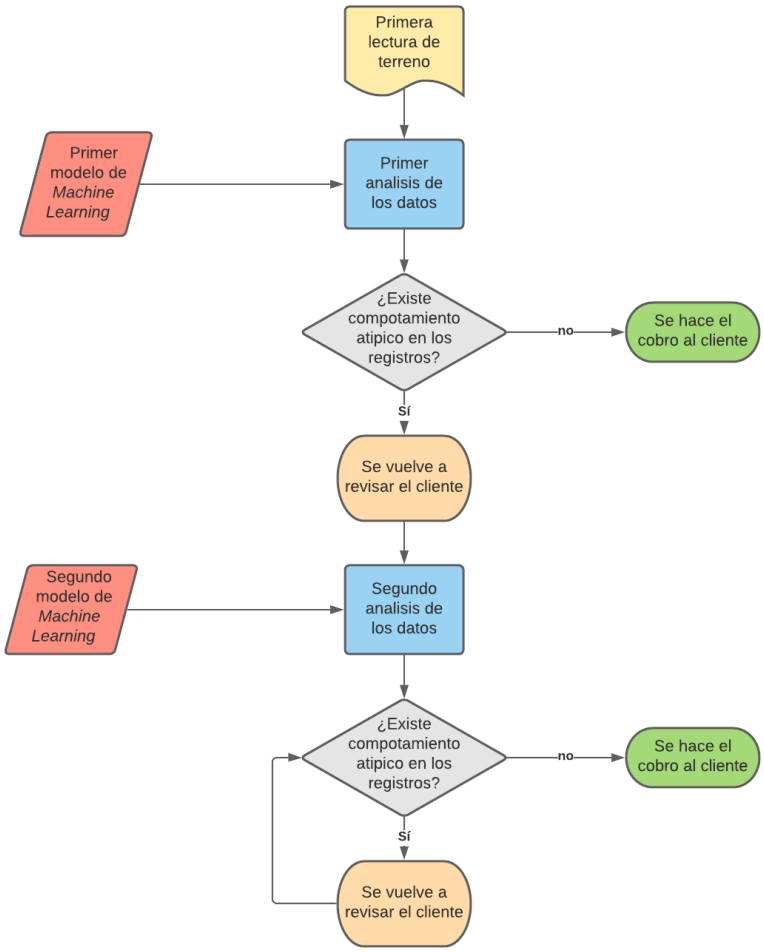

Figura 1-2: Diagrama simplificado de proceso de cobro a clientes incluyendo los modelos de *Machine Learning*.

2 Exploración de datos

En este capítulo se presentará una exploración de los datos con los que se crearán los modelos para finalmente obtener funciones de preprocesamiento de los datos que serán usadas posteriormente.

2.1 Base de datos

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 que serán explicada a continuación:

1. **NRO_SUMINISTRO:** Numero de identificador de cada cliente.
2. **COD_DIAMETRO:** Código del diámetro de cada medidos, el más común en zona residencial es 13. Está relacionado con el flujo de agua que tiene cada medidor, y a su vez con el consumo.
3. **LECTURA:** Lectura final del mes actual que se hace el analista viendo lecturas de varias fuentes.
4. **LECTURA_TERRENO:** Es la primera lectura que hace el lector.
5. **CLAVE_TERRENO:** Especificación de la lectura que hace lector en terreno.
6. **LECTURA_ANT:** Lectura final del mes anterior.
7. **CONSUMO_BASE:** Lectura mes actual menos Lectura mes anterior, este es el consumo del mes actual del cliente.
8. **CONSUMO_PROM:** Promedio de consumo base de 6 periodos anteriores. Puede no ser temporalmente seguidos, ya que puede haber meses sin lectura.
9. **CATEGORIA:** categoría del cliente. Puede ser comercial (C), fiscal (F), institucional (I), Gratuito (G) u otros (O).
10. **TIP_DOCUMENTO:** Tipo del documento del cobro. Puede ser boleta (BO), factura (FA), sin documentos (SD), etc.

11. **COD_LECTOR:** Código del lector, es único para cada lector.
12. **COD_OBS:** Clave que da el lector luego de dar la clave de la lectura. Describe las condiciones del cliente y del medidor. Adjuntas en el Excel "Claves y cod localidad".
13. **COD_LOCALIDAD:** Código de localidad. Las localidades no necesariamente son comunas, son una división que la misma empresa hace según su conveniencia y estudio.
14. **RECORR1:** Sector de facturación. Si el mes se divide en los 20 días hábiles "ideales" cada día se facturas distintos sectores de distintas localidades, todos los sectores que se facturan el mismo día serian un sector de facturación.
15. **RECORR2:** Dentro de las localidades se hace una separación de ellas de manera geográfica de manera que están lo más cerca posible y dentro de un mismo sector de facturación; a esta separación se le llama "Libreta" y esta sería la variable asociada a "RECORR2".
16. **TIE_REMAR:** Tiene remarcador. Se refiere a si el cliente tiene remarcador o no, podría darse el caso en que no se sabe con certeza.
17. **ID_RELACION:** Corresponde a la tarifa.
18. **COD_EMPRESA:** Código de cada empresa. Puede ser ESVAL (1) o Aguas del Valle (2)
19. **CLAVE_LECTURA:** Especificación de la lectura final que hace el lector en terreno, puede diferir de "LECTURA_TERRENO".
20. **CLAVE_TERRENO_MES_ANT:** "CLAVE_TERRENO" del mes anterior.
21. **CONS_BASE_MES_ANT:** "CONSUMO_BASE" del mes anterior.
22. **COD_OBS_MES_ANT:** "COD_OBS" del mes anterior.
23. **CLAVE_TERRENO_MISMO_MES_ANNO_ANT:** "CLAVE_TERRENO" del mismo mes actual, pero del año anterior.
24. **CONS_BASE_MISMO_MES_ANNO_ANT:** "CONSUMO_BASE" del mes actual, pero del mes anterior.
25. **COD_OBS_MISMO_MES_ANNO_ANT:** "COD_OBS" del mes actual, pero del año anterior.
26. **ANNO:** Año actual de la lectura.
27. **MES:** Mes actual de la lectura.
28. **TIENE_REFA**: Etiqueta del modelo. Puede ser: tiene refactura (1) o no tiene refactura (0)

Se nota que la variable que será ocupada como etiqueta de nuestros modelos es “TIENE_REFA”.

2.1.1 Descripción general de los datos

En la *Figura 2-1* se muestra una descripción estadística bastante general de los datos. De esta pequeña descripción se puede rescatar que las desviaciones estándar de los consumos son bastante grandes con lo que se puede concluir que los datos tienen valores bastante irregulares

unos con otros y que es tentativo hacer una subdivisión de estos para mejorar el desempeño de los modelos.

	LECTURA	LECTURA_TERRENO	LECTURA_ANT	CONSUMO_BASE	CONSUMO_PROM	CONS_BASE_MES_ANT	CONS_BASE_MISMO_MES_ANNO_ANT
count	3.034794e+07	3.034794e+07	3.034794e+07	3.034794e+07	3.034794e+07	3.034794e+07	3.034794e+07
mean	1.556831e+03	1.424297e+03	1.622445e+03	1.655484e+01	1.705495e+01	1.633430e+01	1.601982e+01
std	9.466138e+03	9.296046e+03	9.784029e+03	1.561344e+02	1.508750e+02	1.555159e+02	1.519908e+02
min	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	2.700000e+02	1.510000e+02	3.250000e+02	4.000000e+00	6.000000e+00	4.000000e+00	4.000000e+00
50%	9.410000e+02	7.800000e+02	1.012000e+03	1.000000e+01	1.100000e+01	1.000000e+01	1.000000e+01
75%	2.097000e+03	1.942000e+03	2.163000e+03	1.700000e+01	1.700000e+01	1.700000e+01	1.700000e+01
max	4.940572e+06	9.999928e+06	4.878159e+06	7.867700e+04	5.837700e+04	1.078630e+05	1.078630e+05

Figura 2-1: Descripción general de los datos con valores numéricos.

2.1.2 Valores nulos

En la *Figura 2-2* se muestra la cantidad de valores nulos por columna. Se nota que a pesar de que en algunas columnas se llega a los miles de valores nulos no son significativamente demasiados comparados con los datos totales.

```
NRO_SUMINISTRO                      0
COD_DIAMETRO                        1
LECTURA                             0
LECTURA_TERRENO                     0
CLAVE_TERRENO                       0
LECTURA_ANT                         0
CONSUMO_BASE                        1
CONSUMO_PROM                        0
CATEGORIA                        2172
TIP_DOCUMENTO                       0
COD_LECTOR                        935
COD_OBS                           936
COD_LOCALIDAD                       0
RECORR1                             0
RECORR2                             0
TIE_REMAR                     5197537
ID_RELACION                         0
COD_EMPRESA                         0
CLAVE_LECTURA                       0
CLAVE_TERRENO_MES_ANT          756369
CONS_BASE_MES_ANT                   1
COD_OBS_MES_ANT                756854
CLAVE_TERRENO_MISMO_MES_ANNO_ANT    0
CONS_BASE_MISMO_MES_ANNO_ANT   756369
COD_OBS_MISMO_MES_ANNO_ANT     757302
ANNO                                0
MES                                 0
TIENE_REFA                          0
dtype: int64
```

Figura 2-2: Número de valores nulos por columna de los datos.

Para los consumos solo existe 1 valor nulo que se llevara a cero para no eliminar el registro. El resto de los valores nulos es categoría, siendo la variable “TIE_REMAR” la que tiene el mayor número. Lo que se hará con estos valores nulos será cambiarlos por la moda de cada variable, esto se propuso dentro de las reuniones junto con los expertos de la empresa. Así, por ejemplo, los nulos de la variable “CATEGORIA” se llevarán a “R” o los nulos de la variable “TIE_REMAR” se llevarán a “N”.

2.2 Análisis y filtrado de datos atípicos

La *figura 2-3* 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.

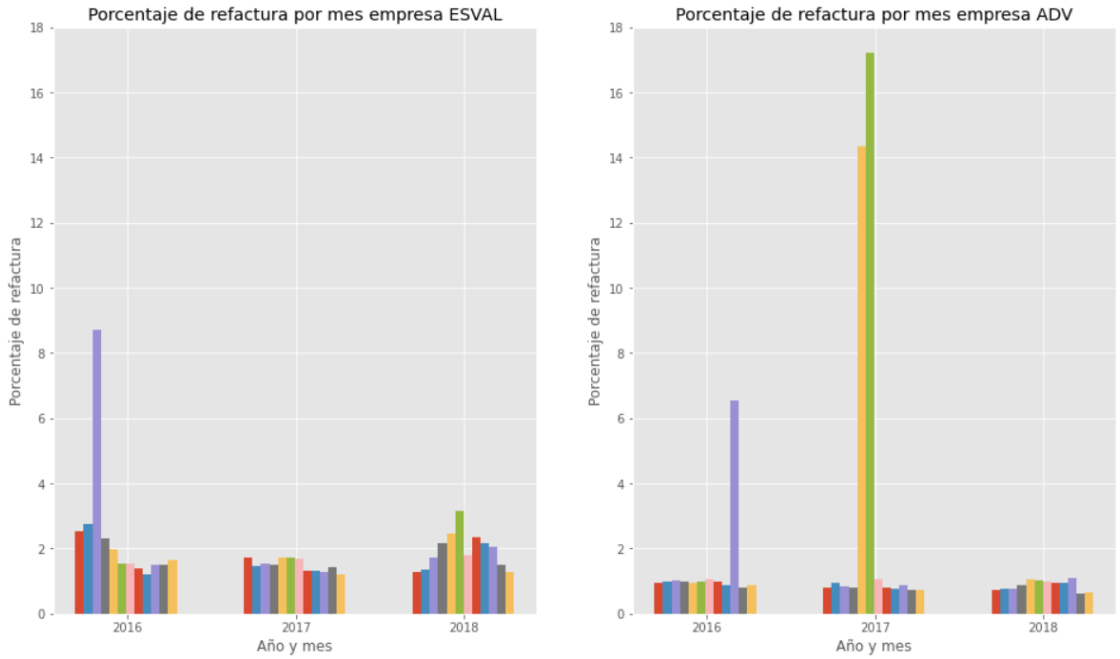

Figura 2-3: Graficas de cantidad de refactura por mes y año.

Para detectar las razones de esta cantidad atípica de refacturas se analizaron graficas hechas en POWER BI para luego realizar un filtrado de cada mes atípico por separado. Las gráficas analizadas se muestran en las *Figuras 2-4, 2-5, 2-6, 2-7* y *2-8*.

Para entender las gráficas de POWER BI, se debe ver que las barras de color más claro corresponden a los datos generales de todos los meses y las barras de color más oscuro corresponden al mes atípico que se está analizando. Luego de ver esto, se procedió a eliminar las categorías que presenta un excedente de refacturas en los meses atípicos.

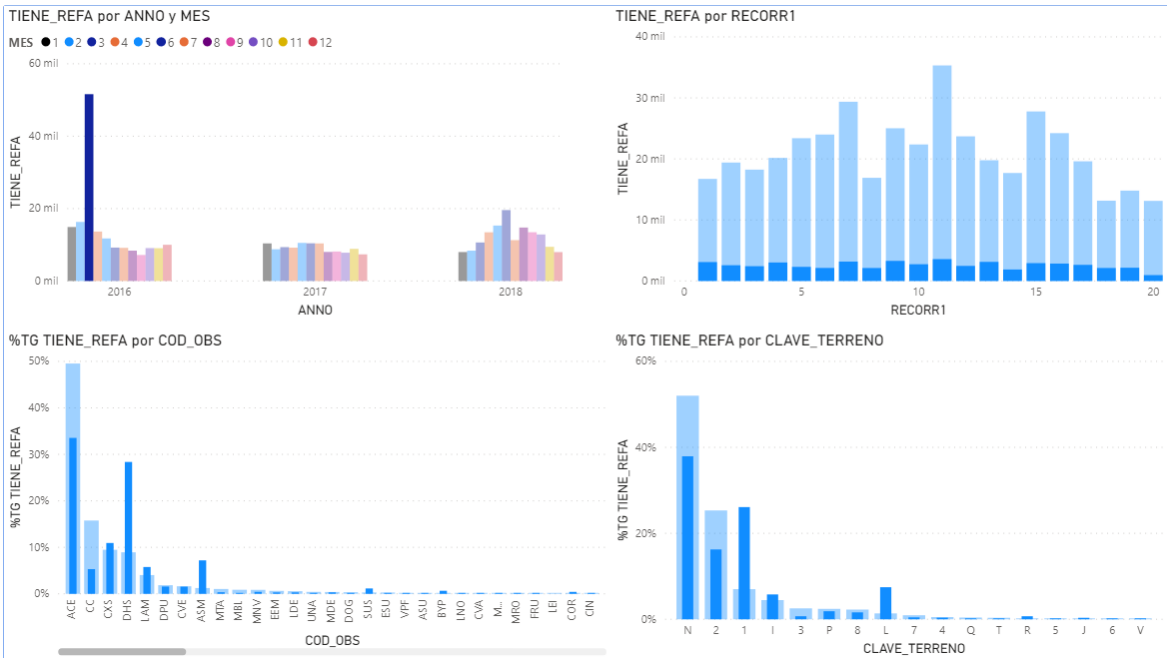

Figura 2-4: Análisis datos atípicos marzo de 2016.

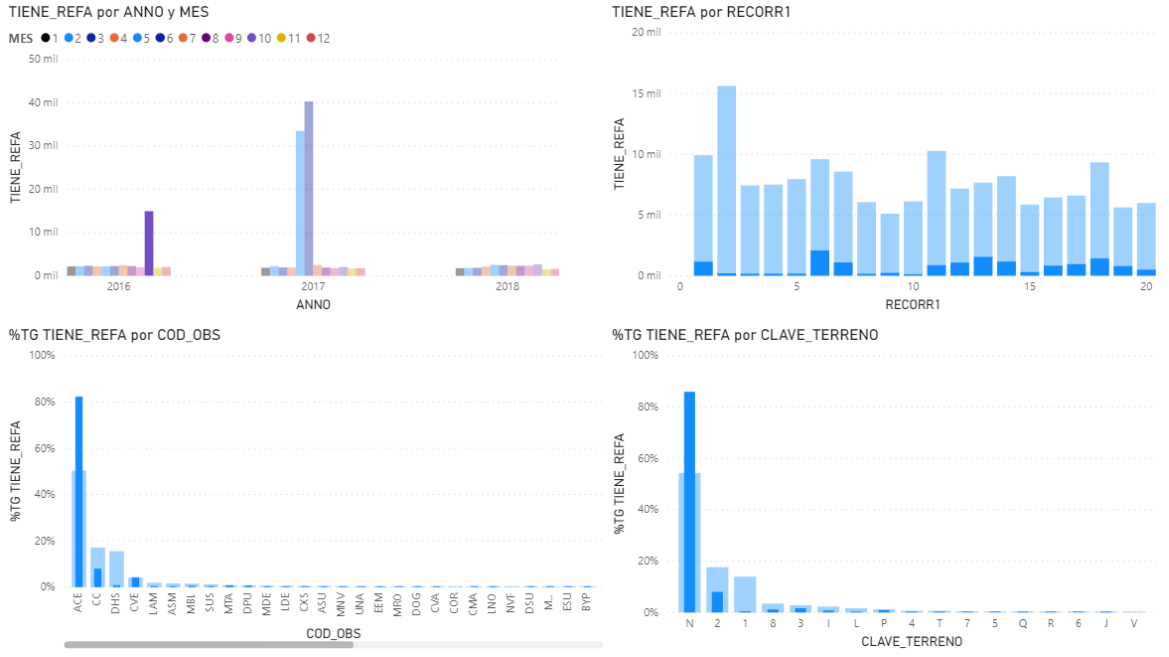

Figura 2-5: Análisis datos atípicos octubre 2016.

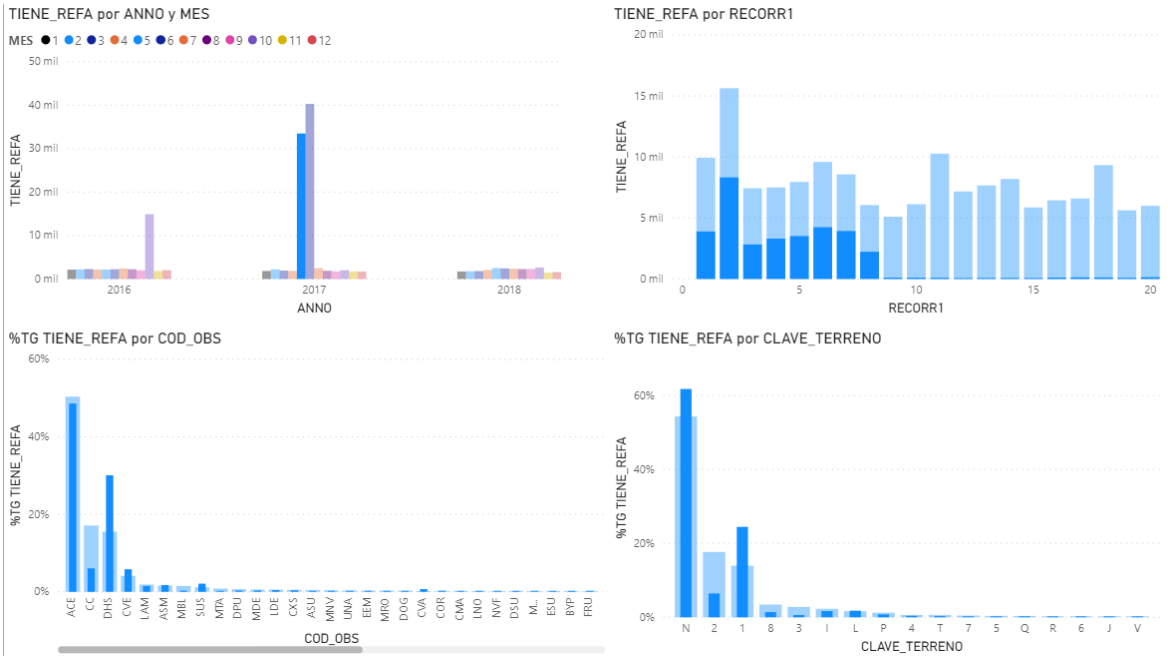

Figura 2-6: Análisis datos atípicos mayo de 2017.

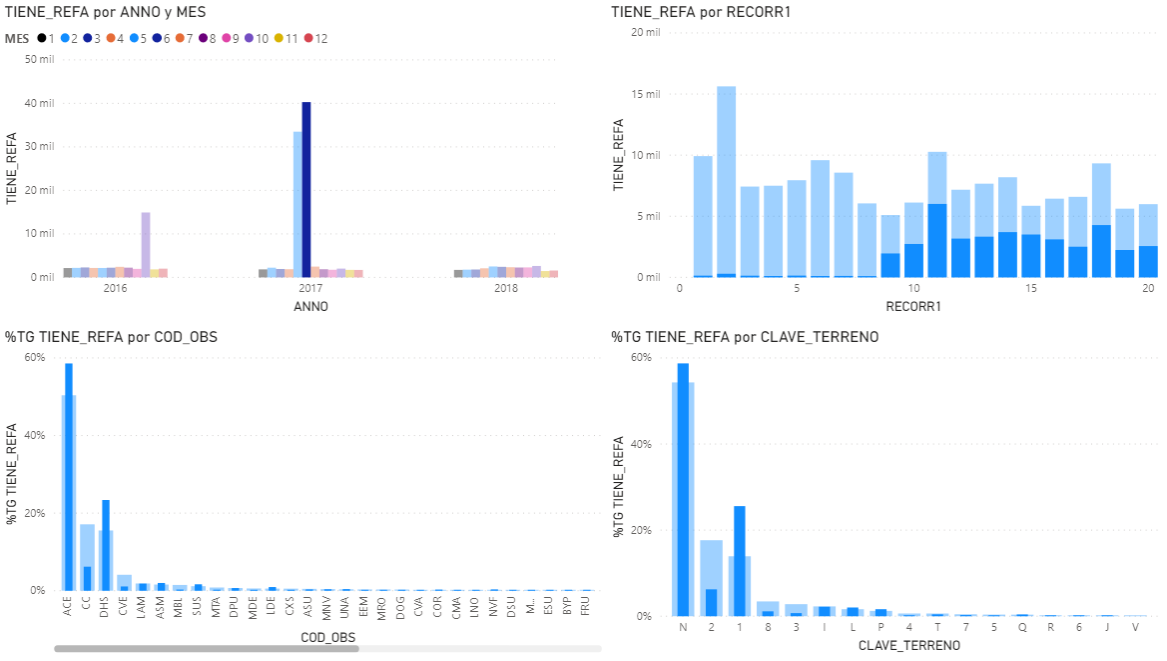

Figura 2-7: Análisis datos atípicos junio de 2017.

Figura 2-8: Análisis datos atípicos junio de 2018.

Luego de analizar las gráficas se filtran los datos durante el preprocesamiento de estos y se llega a una nueva gráfica por mes y año de las refacturas donde se puede ver el cambio de los meses con datos filtrados. Esto se puede ver en la *Figura 2-9*. Se logra que los meses atípicos tengan un comportamiento parecido a los demás meses en el sentido de porcentaje de refactura.

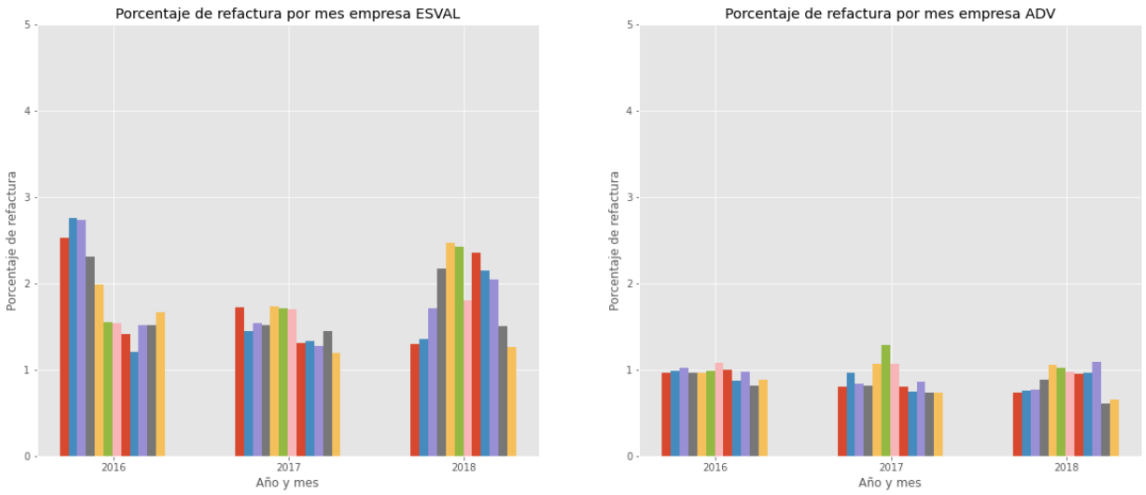

Figura 2-9: Refacturas por mes y años después del filtrado de los datos atípicos.

Cabe destacar que, aunque el filtrado parece tener buenos resultados puede ser una mejor opción simplemente eliminar los datos atípicos. Esta decisión depende de la cantidad de registros que se tiene para entrenar los modelos y de la eficacia de los modelos al comparar los entrenados con datos atípicos filtrados y sin ellos.

2.3 Correlaciones entre columnas

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 2.10*.

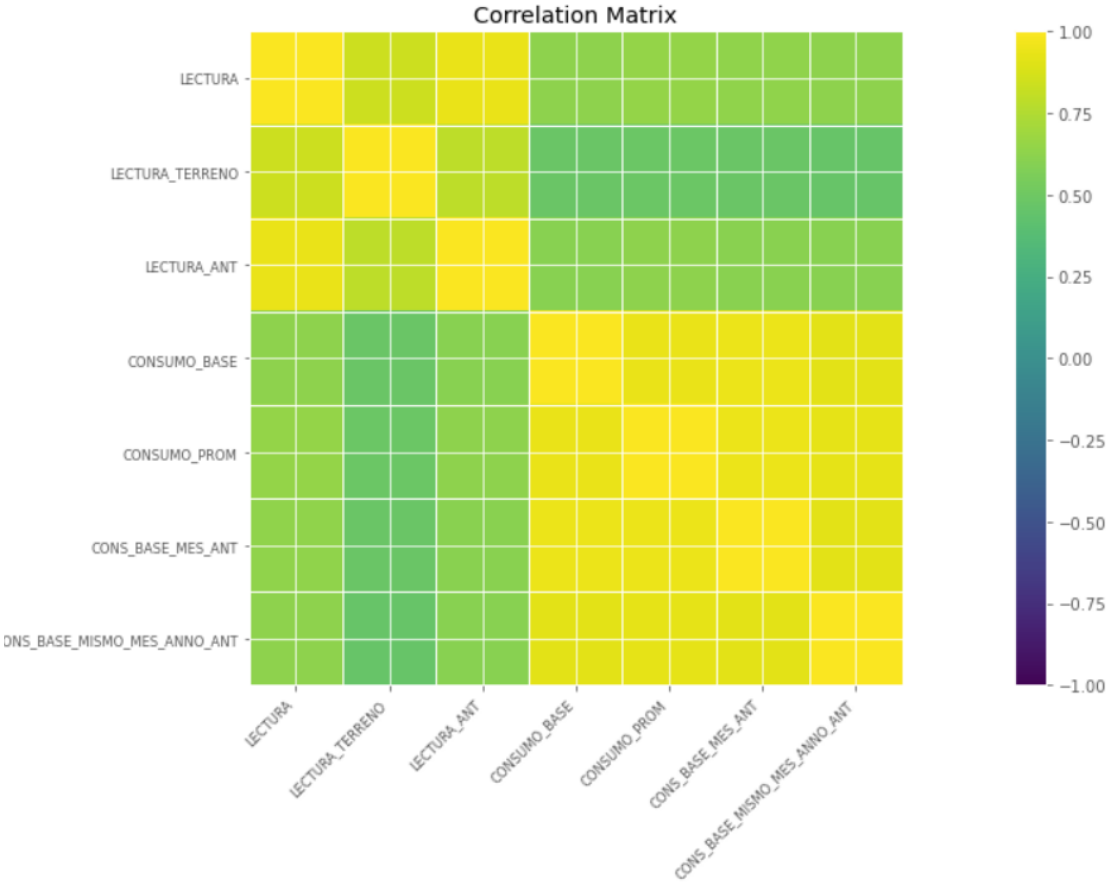

Figura 2-10: Matriz de correlaciones de variables numéricas.

Analizando la matriz de correlaciones se puede notar que existe poca correlación entre las variables a excepción de algunas puntuales. Estas excepciones son "LECTURA" con "LECTURA_ANT" y "LECTURA_TERRENO"; "CONS_TERRENO_MES_ANT" con "CONS_BASE_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.

2.4 Graficas de proporciones de refacturas

La proporción de refacturas por empresa se puede apreciar en la *Figura 2-11.* 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.

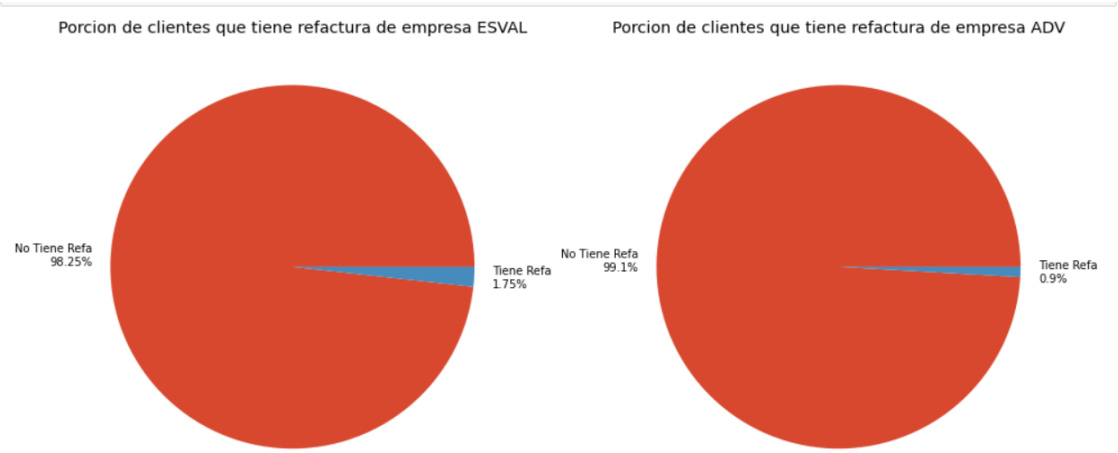

Figura 2-11: Porción de refactura por empresa.

2.4.1 Propoción de refactura por mes

En la *Figura 2-12* 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 sólo 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 3 meses después de ser hecho el reclamo. Por esta razón existe un número más elevado de refacturas hasta el mes de junio.

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

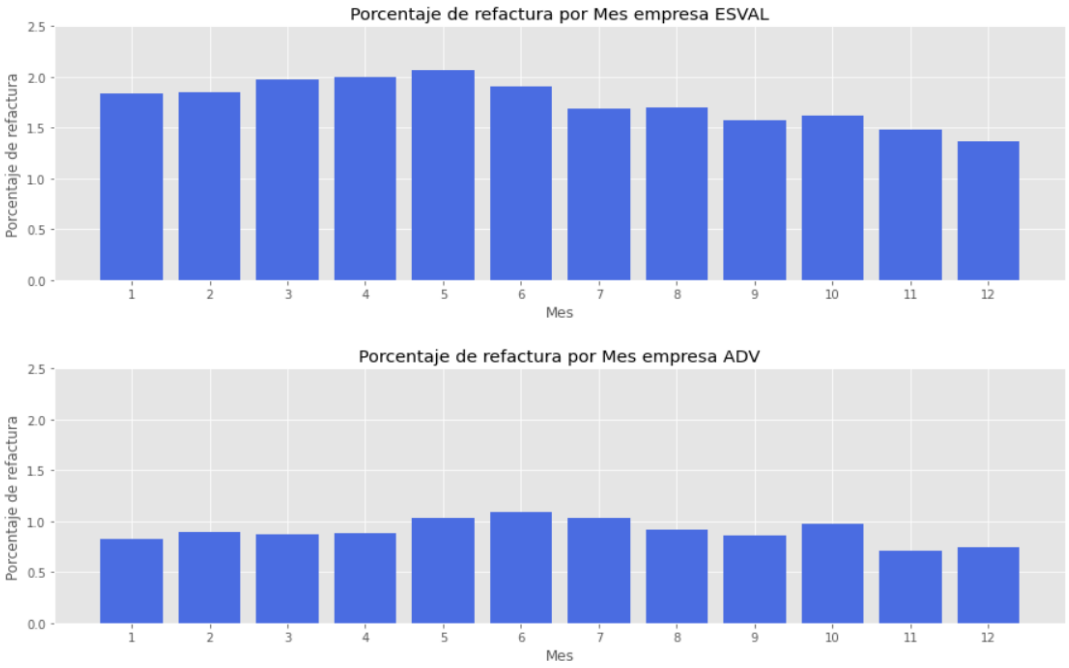

Figura 2-12: Gráficos que muestran las refacturas por mes de cada empresa.

2.4.2 Propoción de refacturas por sector de facturación

La *Figura 2-13* muestra el porcentaje de refacturas por sector de facturación y empresa. Se nota de esta gráfica que la empresa Aguas del valle tiene menor probabilidad de refactura en todos los sectores de facturación y que no existe homogeneidad entre los distintos sectores.

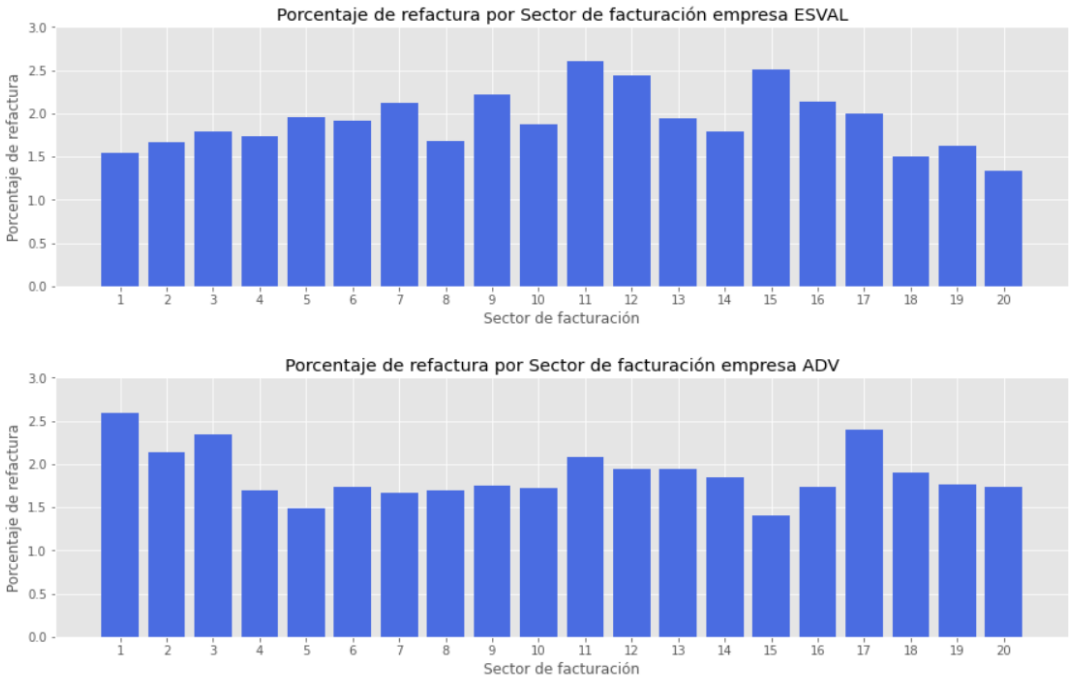

Figura 2-13: Porcentaje de refacturas por sector de facturación y empresa.

2.4.3 Proporción de refacturas por categoría de cliente

La *Figura 2-14* 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 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.

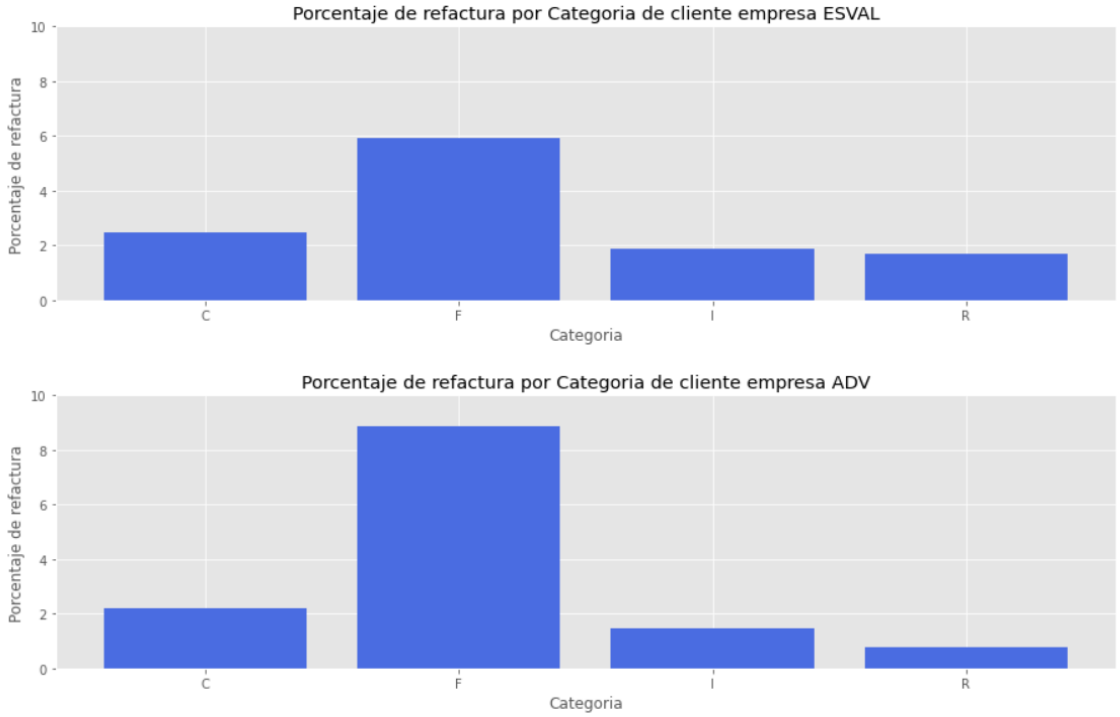

Figura 2-14: Porcentaje de refacturas por categoría de cliente y empresa.

2.4.4 Proporción de refacturas por remarcador

La *Figura 2-15* muestra el porcentaje de refacturas por remarcador y empresa. Se nota de la gráfica que un porcentaje más elevado de refacturas ocurre cuando existe matriz que corresponde al índice "M". Según lo conversado con expertos de la empresa esto se debe a que las matrices están conectadas a varios clientes de, por ejemplo, un edificio. Entonces, cuando existe una refactura debe hacérsele a todos los clientes conectados a esa matriz, lo que hace aumentar el porcentaje de refacturas.

Según los expertos de la empresa el hecho de tener más clientes conectados a matriz influye en la cantidad de refacturas en localidades con más edificios, como lo son: Viña del mar, Con Con o La Serena.

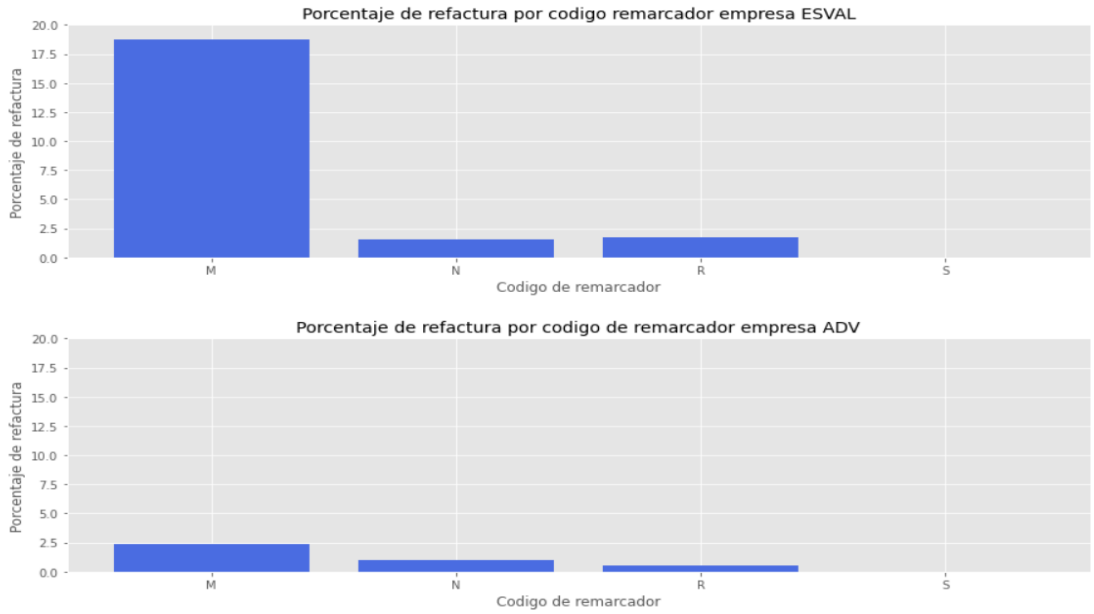

Figura 2-15: Porcentaje de refacturas por código de remarcador y empresa.

2.4.5 Proporción de refacturas por código de diametro

La *Figura 2-16* muestra el porcentaje de refacturas por código de diámetro y por empresa. Se nota de la gráfica que los códigos de diámetros que tienen un mayor porcentaje de refacturas en la empresa ESVAL son “32”, “38”, “50”, “75” y “100”. Mientras que en la empresa Aguas del valle son “32”, “50” y “150”. Lo anterior podría tener explicación en que estos diámetros corresponden a clientes con matrices.

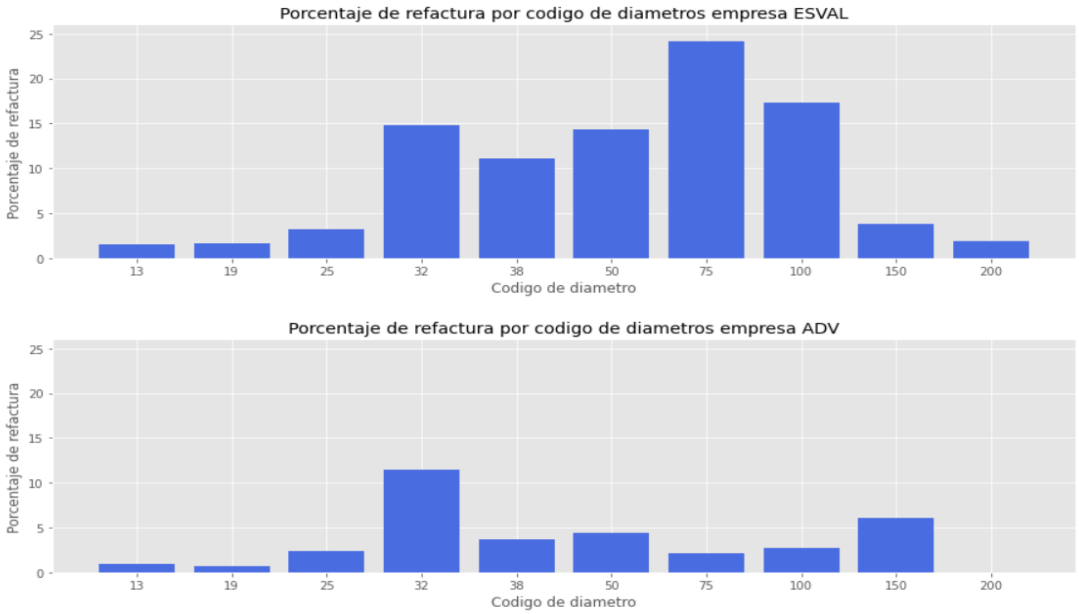

Figura 2-16: Porcentaje de refactura por código de diámetro y empresa.

2.5 Análisis de refacturas por consumos y lecturas

En este subcapítulo se analizarán las refacturas graficando por consumo y por lecturas. Esto se hará para intentar detectar alguna relación entre alguna de estas variables y la incidencia en refactura de los clientes.

2.5.1 Refacturas por consumos

La *Figura 2-17* los consumos de todos los clientes contrastándolos con los consumos de meses anterior y los consumos promedios de cinco meses atrás.

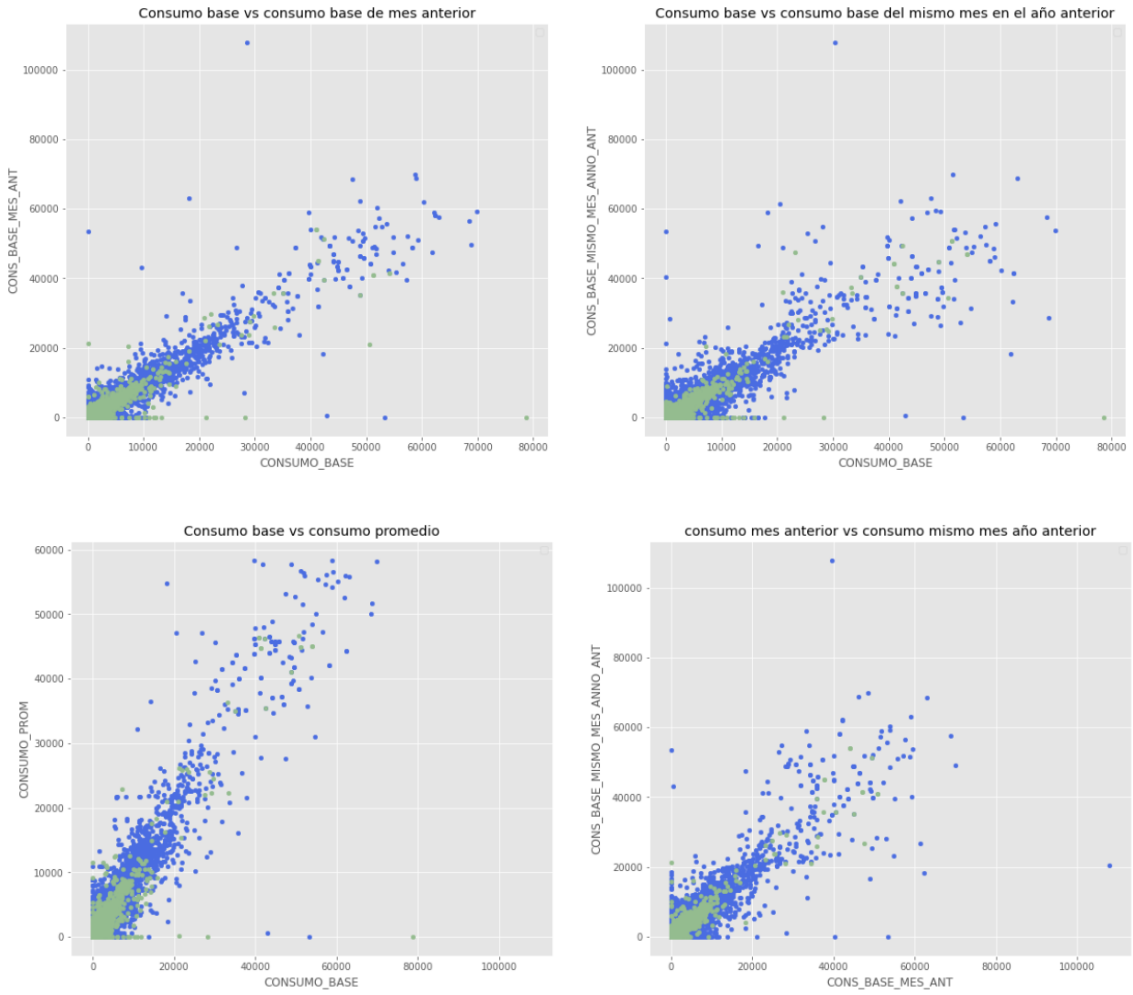

Figura 2-17: Consumos versus distintos otros consumos de clientes con refacturas (verde) y clientes sin refacturas (azul).

De las gráficas anteriores se puede notar un comportamiento bastante lineal, esto quiere decir que en general no se varía el consumo de los clientes entre meses anterior ni de años pasados. Quizá se podría concluir de las gráficas que la mayor cantidad de refacturas están en clientes de bajo consumo, pero que también ahí se concentran la mayor cantidad de clientes sin refactura.

Lo anterior indica que los modelos que se diseñen serán altamente no lineales y podrían descartarse modelos que funcionan mejor en ambientes lineales como SVM (Support Vector Machine [2]).

2.5.2 Refacturas por lectura

La *Figura 2-18* muestra las lecturas de todos los clientes contrastándolos con las lecturas de meses anteriores y las lecturas de terreno.

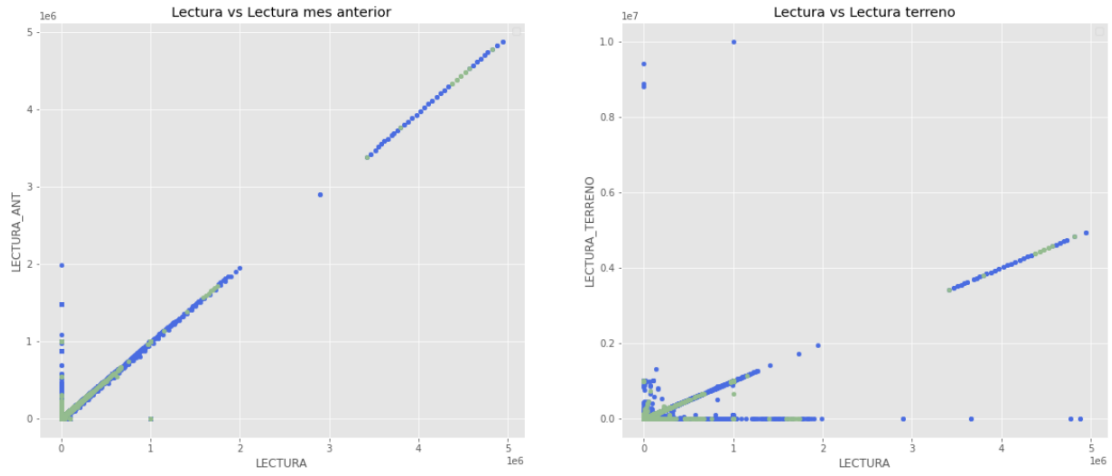

Figura 2-18: Lecturas versus lecturas de mes anterior y lectura terreno de clientes con refacturas (verde) y sin refactura (azul).

De las gráficas se puede notar que el comportamiento es bastante línea salvo en algunos casos donde existen lecturas con valores muy pequeños o cero, estos casos especiales no tienen una relación en cuanto a los clientes con refacturas o no. Solo se podría concluir que las lecturas no tienen mucha incidencia en las refacturas.

2.6 Analisis de componentes pricipales PCA (Principal Component Analysis)

El análisis de componentes principales [3] es una proyección de las distintas variables, que conllevan distintas dimensiones, disminuyendo así las dimensiones hasta un número donde puedan ser graficadas.

En la *Figura 2-19* se muestra un ejemplo sencillo donde se hace un análisis de componentes principales pasando desde 2 dimensiones a 1 dimensión, como se nota en esta grafica este método tiene cierta pérdida de información que no puede ser recuperada. Lo anterior es aclarado porque este método también puede ser ocupado para el preprocesamiento de los datos disminuyendo las dimensiones de los datos de entrada de los algoritmos para aumentar su rendimiento, al ser una base de datos desbalanceada no se ocupará para este propósito porque esa pérdida podría afectar en demasía las variables que inciden para capturar la información de los datos y clasificarlos de buena manera.

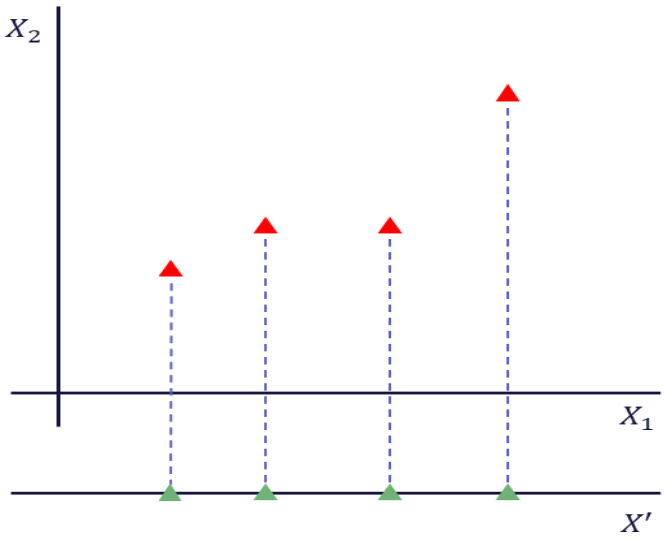

Figura 2-19: Ejemplo sencillo de análisis de componentes principales. Se transforman datos con dos dimensiones (X_1 y X_2) en una dimensión (X'),

En la *Figura 2-20* 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 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.

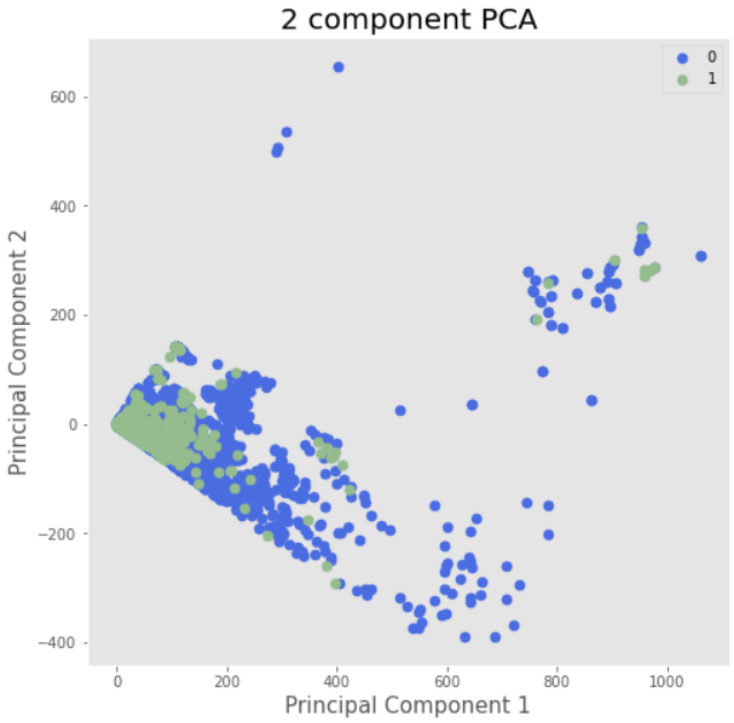

Figura 2-20: Grafica de los 2 componentes principales de los datos.

2.7 Árbol de decisión

El algoritmo de árbol de decisión [4] puede ocuparse para clasificar datos, por esto que estará dentro de los algoritmos que se propondrán en capítulos posteriores. Pero, en este subcapítulo el algoritmo de árbol de decisión se ocupará para dar una descripción aproximada de la importancia de cada variable.

El árbol de decisión toma un registro y según si es mayor o menor a un umbral de ciertas variables decide a que nodo de decisión pasa a continuación, lo que finalmente lleva a las hojas finales donde se decide si el registro tiene refactura o no.

En la *figura 2-21* se muestra un árbol de decisión que fue hecho para fines visuales donde se muestran solo cuatro niveles de este. Se nota que las variables con que primero se decide es "CONS_BASE_MISMO_MES_ANNO_ANT" que corresponde al consumo del mismo mes del año anterior, este se separa en consumos mayores y menores que 4.5, esto quiere decir que esta variable podría ser importante para clasificar de buena manera los datos.

También se debe tomar en cuenta que para las variables categorías se realizó una codificación en caliente (One Hot Encoder) [5], que quiere decir por cada variable categórica se agregan columnas que contienen ceros y unos, ubicando un 1 en el registro que tiene cierta categoría y 0's en todas las demás columnas agregadas. Esto quiere decir que si un nodo de decisión del árbol tiene como condición que sea mayor o menor a 0.5, corresponde a si el registro tiene (mayor) o no tienen (menor) esa categoría.

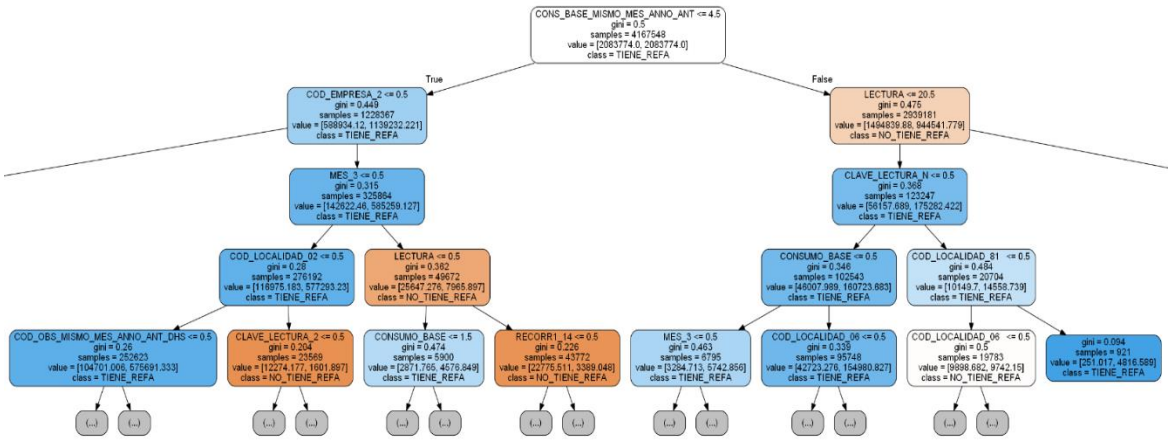

Figura 2-21: Parte de árbol de decisión.

De la figura anterior también se puede notar que muestra una hoja o nodo final después de cuatro niveles, estos niveles son:

1. Si "CONS_BASE_MISMO_MES_ANNO_ANT" menor que 4.5 es falso.

2. Si “LECTURA” menor que 20.5 es verdadero.
3. Si “CLAVE_LECTURA_N” menor que 0.5 es falso.
4. Si “COD_LOCALIDAD_81” menor a 0.5 es falso.

En lenguaje común esto quiere decir que si el consumo base del mismo mes del año anterior es mayor a 4.5, la lectura es menor que 2.5, la clave de lectura es normal (“N”) y la localidad es 81, entonces muy probablemente ese cliente tendrá una refactura, o mejor dicho el modelo clasifica esos datos como clientes con refacturas.

2.8 Datos con y sin responsabilidad del analista

Durante la exploración de los datos se acordó que el modelo será usado en dos instancias, como se dice en capítulo 1 en la *Figura 1-2*. 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. Se debe notar que los datos analista contienen un 6,1% del total de registros de los datos no-analista.

Figura 2-22: Porcentaje de refacturas de datos analista y no-analistas.

La *Figura 2-22* 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 número pequeño de registros con refacturas. Los modelos 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.

3 Preprocesamiento de datos, algoritmos y métricas

En este capítulo se presentarán los algoritmos que se ocuparán para diseñar los modelos, además de dar una pequeña explicación de estos. También se dará un recorrido por los preprocesamientos de los datos que mostraron resultados como los que no, esto para que el lector pueda atestiguar el trabajo de investigación y prueba-error hecho en este trabajo.

3.1 Preprocesamiento de datos

En este subcapítulo se expondrá los preprocesamientos probados para entregar los datos a los algoritmos de aprendizaje y se indicaran los que se ocuparon finalmente.

3.1.1 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* [6], esta función normaliza los datos entre 0 y 1 dependiendo del valor máximo y mínimo de cada variable por separado. La función general para esta codificación se muestra a continuación:

$$minimax = \frac{dato - min}{max - min}$$

Donde “dato” corresponde la variable del registro a normalizar, “max” corresponde al registro de mayor valor de la variable y “min” corresponde al registro de menor valor de la variable.

Para las variables categóricas se hará una codificación en caliente (One Hot Encoder) [5] en la cual se agrega una columna por cada categoría de una variable (esto para todas las variables categóricas). Luego, se rellenan estas 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 3-1* muestra un ejemplo sencillo de esta codificación.

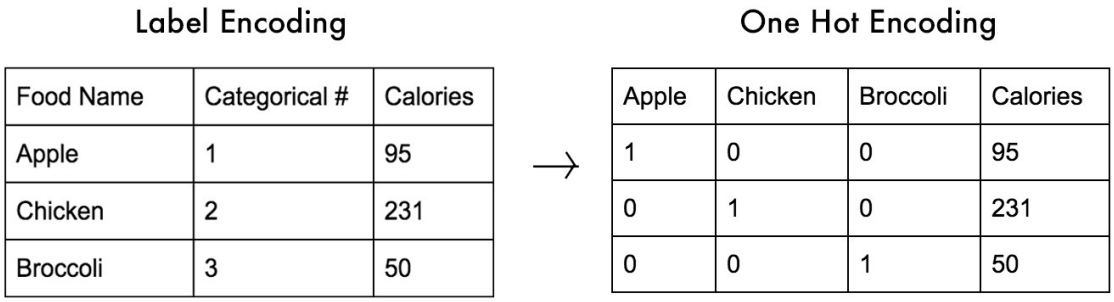

Figura 3-1: Ejemplo sencillo de codificación en caliente (One Hot Encoder)

Luego de esta normalización se hará una normalización estándar, en esta se normalizan los datos ubicando su media en cero. Esto se hará porque el algoritmo de redes neuronales que será explicado en subcapítulos posteriores funciona mejor de esa manera y para los demás algoritmos no tiene mayor relevancia. La función general de esta normalización se muestra a continuación:

$$Normalizacion\ Estandar = \frac{dato - m}{\sigma}$$

Donde "dato" corresponde a la variable del registro a normalizar, "m" corresponde a la media de la variable y "σ" corresponde a la desviación estándar de esta variable.

3.1.2 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.

Como ya se dijo en el subcapítulo 2.8, también se eliminarán las columnas "CLAVE_LECTURA" y "LECTURA" en los datos de no-analista.

3.1.3 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. Por esto se probó con varias divisiones de los datos para comprobar si realmente los modelos se comportan de mejor manera.

Algunas de las divisiones que se probaron fueron por:

- "ID_RELACION", que corresponde a la tarifa de los clientes y separándolos en dos grupos.
- "COD_DIAMETRO", que corresponde a los diámetros de los clientes y separándolos en dos grupos.
- "MES", se dividió en dos temporadas. La primera incluyendo los meses de diciembre, enero, febrero y marzo. La segunda con los demás meses.

Las divisiones anteriores tuvieron peores resultados que los modelos con los datos sin dividir, es por esto por lo que no se ahonda mucho en estos modelos, solo mostraran algunos de sus resultados en capítulos posteriores.

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". Esto explicado para un cliente sería:

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

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 3-2* muestra un histograma con esta nueva columna.

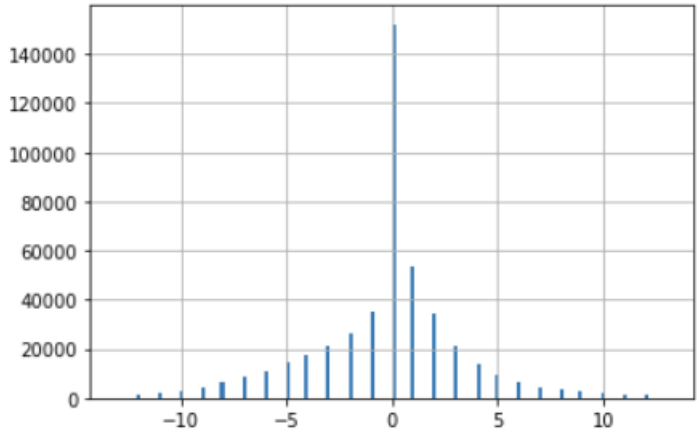

Figura 3-2: Histograma con nueva columna "DIFF_CONSUMO".

3.1.4 Datos atípicos

Como se expuso en el subcapítulo 2.2, ciertos meses de ciertos años con un porcentaje de refactura muy grande comparados con los demás. Estos meses son nombrados como atípicos. Se hizo un análisis de estos meses para crear filtros y así no perder todos los registros de estos meses. En principio los filtros dieron buenos resultados y tenían sentido en cuanto a el porcentaje de refactura de los meses filtrados.

Luego de conversaciones con los expertos de la empresa también se probaron modelos que simplemente eliminaran la totalidad de los registros de los datos de los meses atípicos, atribuyendo esta decisión a que estos datos podrían estar muy sesgados y ser muy heterogéneos con respecto a los demás meses. Lo anterior podría culminar con un peor comportamiento de los algoritmos de aprendizaje. En capítulos posteriores se hace un comparativa de los modelos con datos atípicos filtrados y sin ellos.

3.1.5 Submuestreo y sobremuestro de datos

Existen dos técnicas que son muy usadas en bases de datos desbalanceadas: submuestreo [12] y sobremuestreo [12]. El submuestreo es una técnica en la cual mediante algoritmos matemáticos se eliminan registros de la clase con más de ellos. Por otro lado, el sobremuestreo genera mas registros de la clase con menos de ellos. Estas dos técnicas se basan en proyección o importancia de los datos que eliminan o generan y su fin es hacer que en los modelos disminuya el sobreajuste generado por el desbalance de la base de datos.

Se intentó un submuestreo los datos con las etiquetas que más aparecen en los datos (clientes sin refactura) y sobremuestrear los datos con las etiquetas que menos aparecen (clientes con refactura), esto se hizo porque al ser una base de datos altamente desbalanceada se tiene conocimiento que estos métodos tienen cierta eficacia a la hora de preprocesamiento para modelos de aprendizaje.

Se ocupó la función SMOTETomek [7] para realizar las dos acciones anteriores de manera simultánea. Se noto que no había una mejoría en la clasificación, porque al clasificar más datos con refactura correctamente se mal clasificaban demasiados datos sin refactura y como nuestros modelos dependen de los máximos registros que puede analizar el total de analista se debe tomar en cuenta lo anterior.

3.1.6 Búsqueda de variables importantes con algoritmo Random Forest

La busca de variables importante es método para disminuir la dimensionalidad de los datos y así obtener un mejor resultado de los algoritmos de aprendizaje. Específicamente se ocupó el algoritmo Random Forest [8] que es un algoritmo que crea muchos árboles de decisión de manera aleatoria para testear qué variables son las más o menos ocupadas por estos y para obtener una medida de importancia de cada variable. Se debe tomar en cuenta que, a la hora de calcular los tiempos para el entrenamiento, se debe tomar en cuenta también el tiempo de entrenamiento de este algoritmo dentro del preprocesamiento de los datos.

Se probó esta estrategia y, a pesar de tener bueno resultados generales, cuando se testeo con los sectores de facturación no tubo los resultados esperados. Es por esto por lo que se desistió de este método dejando todas las dimensiones de los datos para los algoritmos de aprendizaje.

3.2 Algoritmos de aprendizaje de maquina

En este subcapítulo se presentarán los cuatro algoritmos de machine learning que se diseñaron para intentar solucionar la problemática. Se hará una breve explicación de cada uno de ellos.

3.2.1 Regresión logística

La regresión logística [8] 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.

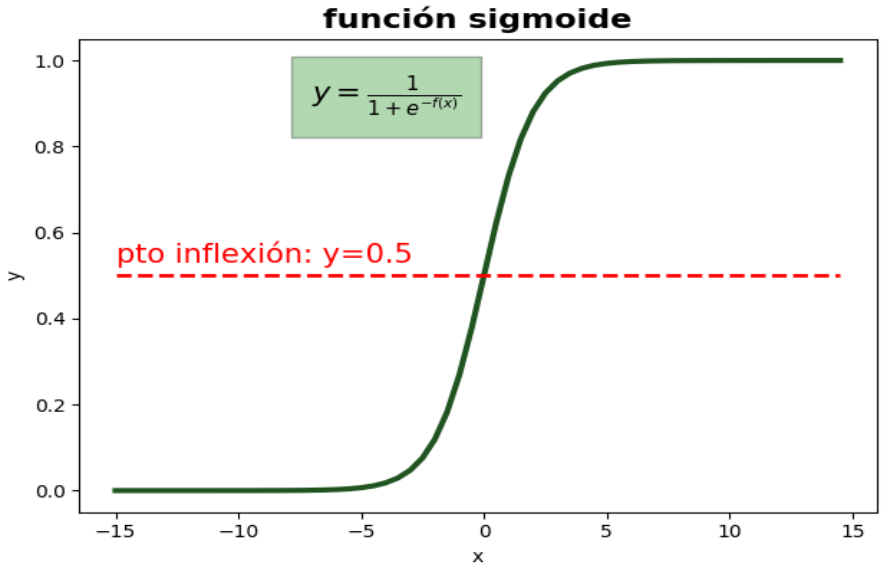

Figura 3-3: grafica de la función sigmoide.

En la *Figura 3-3* se muestra la función sigmoide con la cual se calcula el umbral de decisión, se nota que en gran medida el hecho de ocupar esta función hace que este algoritmo puede capturar características no lineales de los datos.

Los hiper-parámetros que se pueden modificar en este algoritmo son el número de iteraciones y la lambda de la regularización. Esta regularización sirve para que los parámetros no tomen valores muy grandes, y así evitar el sobreajuste, esto es muy deseable con nuestros modelos, ya que al ser una base de datos desbalanceada.

3.2.2 Árbol de decisión

El algoritmo de árbol de decisión [4] 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 por qué variables hacerlo.

En el capítulo 2-7 se mostró un árbol para ejemplificar las variables importantes, en este caso no tiene mucho sentido exponer el árbol ya que este tendrá varias hojas de profundidad y se anchura.

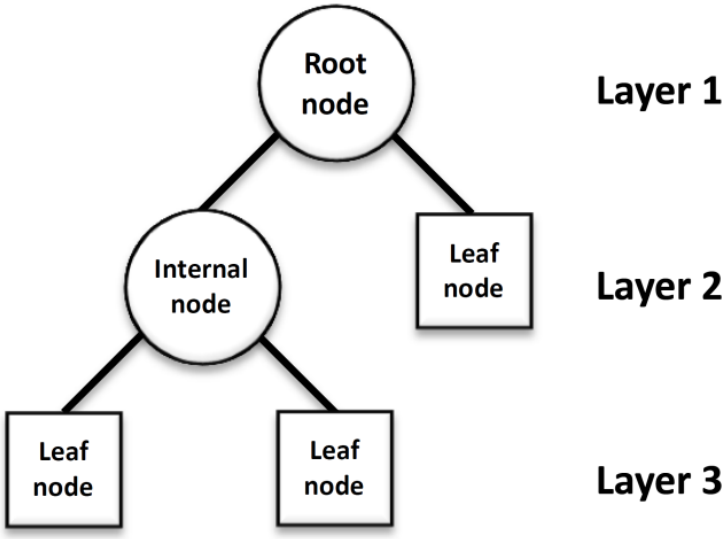

Figura 3-4: Diagrama simplificado de un árbol de decisión.

La *Figura 3-4* muestra un diagrama simplificado se un árbol de decisión. Algunos de los hiper parámetros que pueden modificar en este algoritmo son la cantidad máxima de nodos o la profundidad, la cantidad de hojas o nodos finales, la cantidad de datos necesarios para que un nodo haga una separación.

3.2.3 Redes neuronales artificiales

Las redes neuronales artificiales [9] 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.

Cada neurona de este algoritmo cuenta con pesos que se multiplican con cada variable de la capa anterior para luego pasar por una función de activación. Esta función de activación generalmente es la sigmoide al igual que en la regresión logística, pero puede tener variaciones como la función RELU o ELU.

En la *Figura 3-5* se puede apreciar un diagrama simplificado de un modelo de redes neuronales. Los hiper parámetros que se pueden modificar en este algoritmo son la regularización, el peso de cada clase, la función de decisión de las neuronas, el número de capas y el número de neuronas por capas.

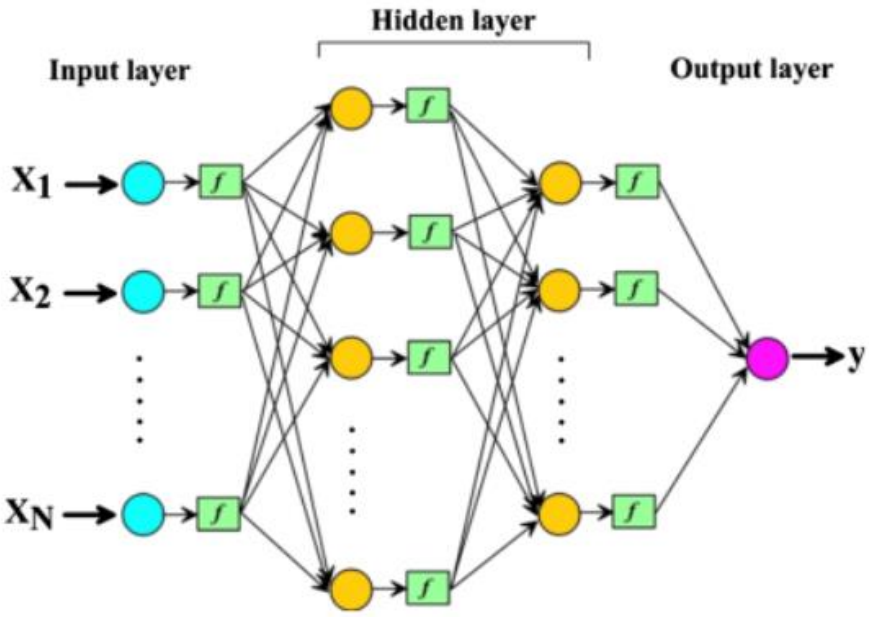

Figura 3-5: Diagrama simplificado de algoritmo de redes neuronales artificiales.

3.2.4 XGBoost

XGBoost [10] es un algoritmo de boosting, 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.

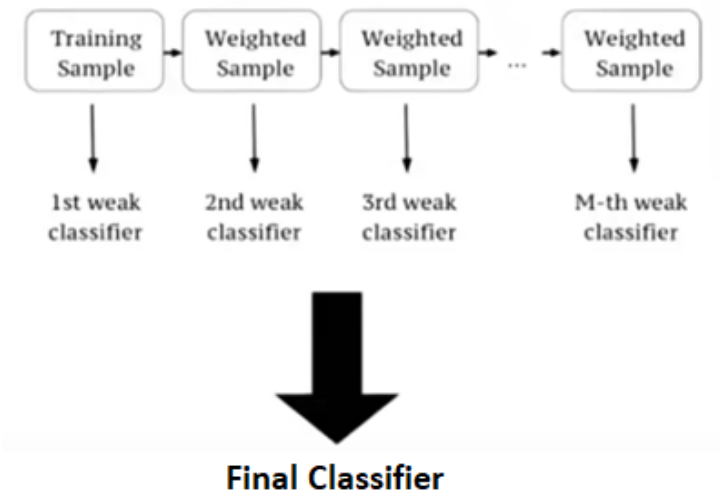

Figura 3-6: Diagrama simplificado de algoritmo XGBoost.

La *Figura 3-6* muestra un diagrama simplificado de este algoritmo. Los hiper parámetros que pueden modificarse para que este algoritmo tenga un mejor desempeño son el peso de cada clase, el valor del parámetro de regularización, el número de árboles a crear, el número máximo de profundidad de cada árbol, el número máximo de hojas o nodos finales y la función objetivo.

3.3 Métricas de evaluación

En este subcapítulo se explicarán las métricas de evaluación que se tomaron para los modelos diseñados y el porqué de estas.

Todas las métricas nacen de la matriz de confusión [11]. Esta matriz tiene como elementos las clasificaciones del algoritmo, estos elementos son:

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

La *Figura 3-7* muestra una matriz de confusión para el caso de clasificación binaria, como es nuestro caso.

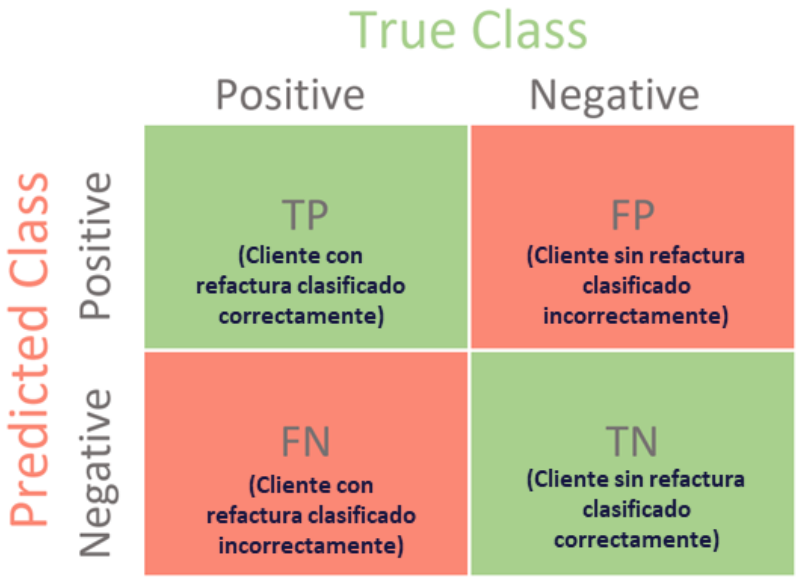

Figura 3-7: Matriz de confusión para clasificación binaria.

Una métrica de evaluación muy usada es el *accuracy*, [11] que se define con la formula siguiente:

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN}$$

Se nota que a pesar de que esta métrica es muy ocupada, no es muy recomendable para bases de datos desbalanceados. Esto porque nuestra base de datos tiene bajo el 2% de datos positivos, por lo que solo clasificándolos todo negativo se lograría un *accuracy* de más de 98%. Lo anterior podría ser engañoso a la hora de comparar los modelos y decidir cuál es mejor.

A continuación, se presentan las métricas de evaluación que se ocuparon para los modelos.

3.3.1 Recall y Precision

El *recall* [11] y la *precisión* [11] 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 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.

3.3.2 F1-score y G-mean

Estas son métricas que forman un promedio de las métricas anteriores. F1-score 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 [11] 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.

3.3.3 F-measure

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* [11] 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 mas importancia a la *precision* (es decir, no importa que haya varios negativos mal clasificados, mientras que los positivos estén bien clasificados).

3.3.4 Curva ROC y metrica AUC

Otra métrica para mejorar las falencias de las métricas anteriores es AUC (*Area Under Curve*) [11], esta depende de la curva ROC [11]. 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.

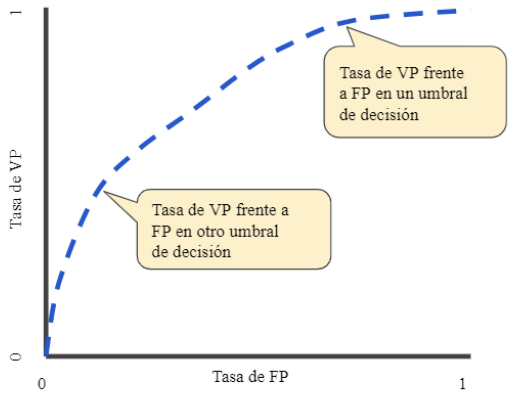

Figura 3-8: Ejemplo de curva ROC.

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

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

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

La *Figura 3-8* muestra un ejemplo de la curva ROC, mientras que la *Figura 3-9* muestra un ejemplo del área que sería nuestra métrica AUC.

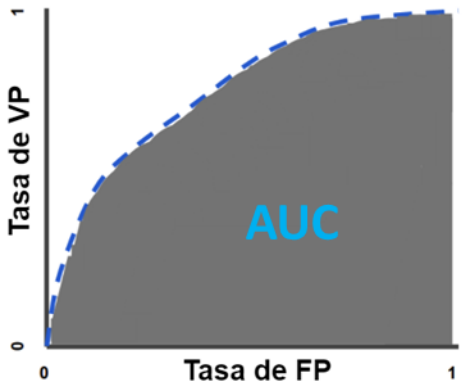

Figura 3-9: Ejemplo de métrica AUC.

3.3.5 AUCM

La métrica AUCM (*Area Under Curve Modified*) es una métrica inventada para el propósito de este trabajo. Esta métrica crea una curva ROC modificada tomando en cuenta el porcentaje de refacturas revisadas en contraste de los datos sin refacturas que se deben revisar con todos los umbrales de decisión.

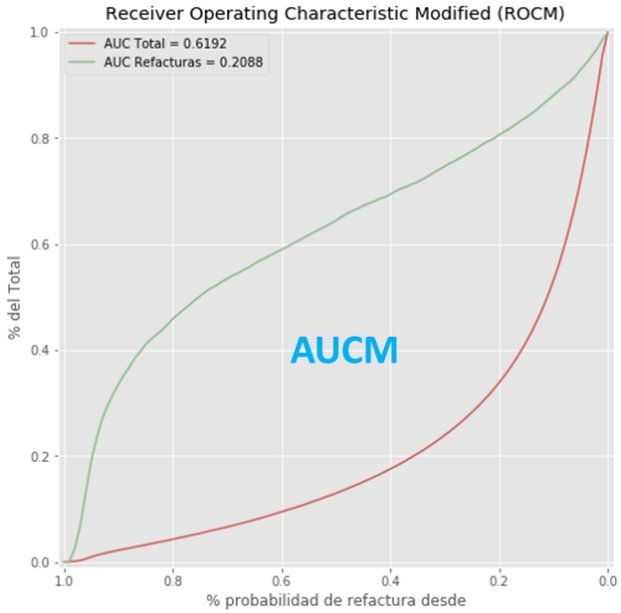

Figura 3-10: Ejemplo de grafica de curva ROCM y métrica AUCM.

La *Figura 3-10* 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. Mientras más cercana a 1 sea esta métrica significa que el modelo tiene mejor rendimiento, esto porque se revisan un mayor porcentaje de clientes con

refactura (verdaderos positivos) y pocos clientes que no la tienen (falsos positivos). Lo peores resultados de esta métrica son valores que se acercan a 0 o que son negativos, ya que se revisarían una gran cantidad de clientes sin refacturas, dejando pocos clientes con refactura dentro de los registros revisados.

La *Figura 3-11* muestra que elementos de la matriz de confusión toma cada curva de esta métrica,

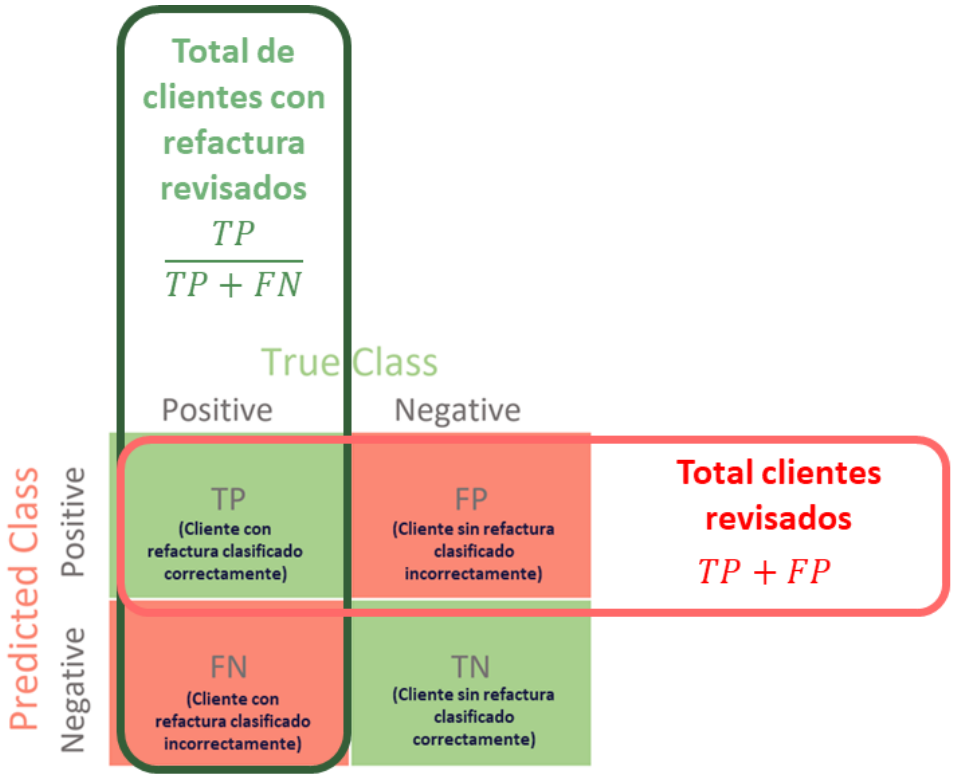

Figura 3-11: Diagrama explicativo de cada curva de la métrica AUCM.

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.

4 Modelos, resultados y tiempos

En este capítulo se presentarán los distintos modelos con distintas divisiones de los datos evaluándolos con las métricas anteriores en el mes de diciembre del año 2018.

4.1 Modelos datos no-analista

En este subcapítulo se presentarán todos los modelos diseñados de los datos no-analista independiente de que tan buenos resultados tengan. Se hará un recorrido temporal por la manera en que se evaluaron y las divisiones de datos que hicieron en los modelos.

Cabe destacar que todos los algoritmos primero pasaron por una fase de búsqueda de hiper parámetros, dentro de esta fase se buscó los mejores hiper parámetros general para poder probar todo estos con el mejor rendimiento posible. Esta fase se repetirá cuando se dé con el mejor preprocesamiento o división de los datos para así hacer un ajuste fino de los modelos.

4.1.1 Modelos sin divisiones con y sin datos atípicos filtrados

Como se dijo en capítulos anteriores se hizo un filtrado de los meses atípicos que parecía tener buenos resultados, esto se analizó diseñando modelos con y sin datos atípicos filtrados y comparando los resultados de ambos antes de avanzar a otra divisiones o preprocesamiento de los datos.

Los resultados de los dos modelos son mostrados en la *Tabla 4-1*. En esta tabla se promediaron los resultados para todos los sectores de facturación, esto se hizo para poder comparar los modelos más fácilmente. Se nota que los mejores resultados en todas las métricas las tienen los modelos sin los datos atípicos filtrados, obteniendo los mejores resultados los modelos de los algoritmos con redes neuronales y de XGboost.

Es por lo anterior que se desiste de ocupar los datos con los meses atípicos filtrados y se enfoca el esfuerzo en buscar el modelo con mejor rendimiento con los datos sin los meses atípicos. Esto se hace porque se debe acotar el diseño de modelos para el estudio, ya que la búsqueda heurística del mejor modelo podría extenderse infinitamente si no es acotada.

Tabla 4-1: Resultados de modelos con y sin datos atípicos

Modelos	F1-score	G-mean	F-measure	AUCM	Porcentaje de refacturas revisadas
LR sin atípicos	0,06813	0,73277	0,650685	0,31941	0,47849
DT sin atípicos	0,065785	0,745445	0,673345	**0,397995**	0,42707
NN sin atípicos	0,10587	**0,75179**	**0,68664**	0,388445	**0,533805**
XGB sin atípicos	**0,114915**	0,66629	0,548015	0,35829	0,485605
LR con atípicos	0,06433	0,719355	0,62678	0,30949	0,46407
DT con atípicos	0,059255	0,729485	0,64664	0,380985	0,42172
NN con atípicos	0,094305	0,739805	0,6622	0,35591	0,514815
XGB con atípicos	0,107235	0,65953	0,53586	0,34493	0,464505

4.1.2 Modelos con divisiones y sin datos atípico

Para encontrar una división de los datos que fuera provechosa para el desempeño de los algoritmos se hizo un estudio de prueba y error con las distintas divisiones. Estas divisiones fueron antes discutidas con los expertos de la empresa para comprobar que tuvieran sentido. Algunas de divisiones probadas, pero que no tuvieron buenos resultados fueron:

- División por categoría de cliente.
- División por empresa.
- División por mes.
- División por tipo de remarcador.
- División por tarifa.
- División por localidad.

Todas las divisiones anteriores no mostraron una mejora con respecto a los modelos sin divisiones en los datos. En la *Tabla 4-2* se muestran los resultados de la división por localidad a modo de ejemplo. Se nota que solo se diseñaron modelos de redes neuronales y del algoritmo XGboost, esto porque como se vio en la *tabla 4-1* estos son los algoritmos con mejor desempeño y se acotara el estudio a estos. Lo anterior fue comprobado con varias pruebas que se hicieron con las divisiones y el mejor resultado fue constante.

Es por todo lo anterior que se desistió de probar modelos con estas divisiones de datos y se optó por unas que dieran mejores resultados.

Tabla 4-2: Resultado de modelos con división de datos por localidad y sin división.

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 con división	0,093	0,72	0,63	**0,39**	0,50
XGB con división	0,08	0,67	0,56	0,34	0,45

Modelos con división por temporada

Según lo hablado con los expertos de la empresa una división que tiene mucho sentido es la por temporada. Esto se nombra en el subcapítulo 3.1.3, donde se definió una nueva columna "DIFF_CONSUMOS" y en la *figura 3-2* se presenta un histograma de esta nueva columna. Se nota que existen 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 subcapítulo 3.1.3, 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. En la *Tabla 4-4* se pueden apreciar los resultados de estos modelos por localidad, mientras que en la *Tabla 4-3* se pueden apreciar los resultados de los modelos sin división de los datos. 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.

Lo anterior puede traer problemas respecto a los tiempos de entrenamiento, ya que estos modelos están pensados en ser reentrenados cada semana esperando que logran terminar su

entrenamiento en los días del fin de semana. Si los tiempos de entrenamiento sobrepasan estos días se optará por acotar los modelos a ocupar.

Tabla 4-3: 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**

En la *Tabla 4-5* se muestran los resultados de los modelos promediados para tener una idea general de que modelos son mejores que otros. A pesar de que se dice en el párrafo anterior, también se nota que a medida que los modelos tienen un mejor desempeño, en general lo hacen para todos los sectores a la vez. Esto es esperable, ya que los sectores de facturación solo entregan información sobre el día en que se hacen las lecturas en terreno y no sobre los patrones que puede tener el lector, la empresa, la localidad o la temporada.

La tabla anteriormente nombrada muestra que los modelos con división por temporada tienen un mejor desempeño que los modelos sin división. Y, por sobre todos los modelos el algoritmo de redes neuronales tiene el mejor desempeño.

Tabla 4-4: 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		0,46		**0,55**
Máximo		0,66		**0,75**
Mínimo		0,32		**0,38**

Tabla 4-5: Comparación de promedios de resultados de modelos con división por temporada y sin división.

Modelos	F1-score	G-mean	F-measure	AUCM	Porcentaje de refacturas revisadas
NN sin división	0,105	0,75	0,68	0,38	0,53
XGB sin división	**0,11**	0,66	0,54	0,35	0,48
NN con división	0,100	**0,76**	**0,69**	**0,40**	**0,55**
XGB con división	0,08	0,69	0,58	0,37	0,46

Modelos con división por código de diámetro

Del análisis hecho del problema por parte del investigador 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".

La *Tabla 4-6* 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 4-6: 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	**0,5**1	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**

De la *Tabla 4-6* se puede notar que los mejores resultados se obtienen con el modelo de redes neuronales, mas no en todos los sectores.

La *tabla 4-7* 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 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 4-7: 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	0,1	0,71	0,63	0,39	0,49

4.2 Modelos datos analista

En este subcapítulo se expondrán los resultados y mejores modelos con los datos con responsabilidad del analista. Los resultados de los modelos con estos datos son bastante similares a los resultados del subcapítulo anterior 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.

Los modelos base son con los datos sin dividir, en la *Tabla 4-8* se muestran los resultados de estos modelos 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.

Tabla 4-8: 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**

Modelos con división por temporada

La *Tabla 4-9* 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 compararlos 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 4-9: 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	0,7	0,36	**0,75**	**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**	**0,46**	0,73	0,46
15	0,71	0,37	**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	0,29	**0,69**	**0,44**
19	**0,80**	**0,34**	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

Modelos con división por diámetro

La *Tabla 4-10* 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.

Cabe destacar, que al igual que en los modelos con división por temporada, para obtener las métricas se separaron los datos por las divisiones, luego se ocupó el modelo especifico de cada división para evaluar los clientes, para finalmente volver a unir los datos y obtener el resultado general de cada sector de facturación.

Tabla 4-10: 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	**0,8**	**0,40**	0,76	0,44
20	0,67	0,36	**0,84**	**0,43**
Promedio	0,681	0,37	**0,77**	**0,45**
Máximo	0,93	0,47	**0,99**	**0,55**
Mínimo	**0,55**	0,27	0,52	**0,35**

Finalmente, en la *Tabla 4-11* 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 4-11: 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	0,080	0,74	0,694	0,388
XGB sin división	0,086	0,71	0,62	0,34
NN por temporada	0,09	0,72	0,65	0,387
XGB por temporada	0,06	0,742	0,69	0,33
NN por diámetro	**0,10**	**0,791**	**0,77**	**0,45**
XGB por diámetro	0,07	0,741	0,68	0,37

4.3 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. 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.

Tabla 4-12: 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

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.

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 4-13: 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

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.

Discusión y 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.

Trabajos y propuestas futuras

Según los analizado en este trabajo y lo 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 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 sólo 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.

Bibliografía

[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] Cortes, C., & Vapnik, V. (1995). Support vector machine. *Machine learning*, *20*(3), 273-297.

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

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

[5] 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.

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

[7] Wang, Z., Wu, C., Zheng, K., Niu, X., & Wang, X. (2019). SMOTETomek-Based Resampling for Personality Recognition. *IEEE Access*, *7*, 129678-129689.

[8] Biau, G. (2012). Analysis of a random forests model. *The Journal of Machine Learning Research*, *13*(1), 1063-1095.

[9] Jovell, A. J. (1995). *Análisis de regresión logística*. Madrid: Centro de Investigaciones Sociológicas.

[10] 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.

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

[12] Hernandez, J., Carrasco-Ochoa, J. A., & Martínez-Trinidad, J. F. (2013, November). An empirical study of oversampling and undersampling for instance selection methods on imbalance datasets. In *Iberoamerican Congress on Pattern Recognition* (pp. 262-269). Springer, Berlin, Heidelberg.