UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FíSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

SISTEMA DE ANÁLISIS DE TÓPICOS PARA INTERACCIONES CLIENTE-CALL CENTER

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELÉCTRICO

FRANCISCO IGNACIO NICOLAI MANAUT

PROFESOR GUÍA:
CLAUDIO ANDRÉS GARRETÓN VÉNDER

MIEMBROS DE LA COMISIÓN:
MARCOS ORCHARD CONCHA
RICHARD WEBER HAAS

Este trabajo ha sido parcialmente financiado por Entel S.A.

SANTIAGO DE CHILE
2019

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERA CIVIL ELÉCTRICA
POR: FRANCISCO IGNACIO NICOLAI MANAUT
FECHA: 2019
PROF. GUÍA: CLAUDIO ANDRÉS GARRETÓN VÉNDER

SISTEMA DE ANÁLISIS DE TÓPICOS PARA INTERACCIONES CLIENTE-CALL CENTER

Actualmente vivimos en una era digital donde tanto la generación de datos como las capacidades computacionales evolucionan exponencialmente. Consecuentemente, muchos servicios nuevos surgen mientras que otros ven redefinida la forma en que se entregan. Un caso específico de ello son los call center, donde potenciales automatizaciones no solo representan reducción de costos para empresas, sino que también un mejor desempeño en su quehacer.

Con el objetivo de identificar de mejor forma la razón de llamada de los clientes al call center de los servicios móviles de Entel, se construye un sistema capaz de recibir breves transcripciones de las llamadas para determinar el motivo de las mismas. Dicho sistema se divide en dos componentes, una dedicada a clientes con suscripción y otra dedicada a clientes de prepago.

Para ello, se estudiaron diversas técnicas y formas de preprocesar los datos de entrada, optando finalmente por una configuración con remoción de caracteres inválidos, corrección de ortografía, uso de listas de stopwords, generación de n-gramas, lematización y stemming.

En primera instancia, el sistema propuesto consistió en el desarrollo de un clasificador basado en el modelo Latent Dirichlet Allocation. A pesar de que dicho esquema presentó resultados positivos en desempeño. El desarrollo y uso de este se mostró difícil de sostener en el tiempo puesto que el ejercicio de traducir la salida del modelo LDA a una etiqueta legible es poco estable.

En consecuencia, se propuso e implementó un nuevo sistema con un modelo supervisado que recibe como entrada el vector entero que el modelo LDA anteriormente ajustado retornaba. Con esta implementación y gracias a la adición de datos categóricos asociados a la acción a realizar para solucionar el problema por el cual llaman los clientes, se obtuvieron fuertes mejorías en el desempeño global del sistema.

Finalmente, el sistema desarrollado tuvo un desempeño en Fscore ponderado de 0,78 para la base de clientes suscritos y de 0,88 para la base de clientes de prepago. La principal causa tras el menor desempeño en suscritos corresponde al fuerte desbalance que hay entre los motivos de llamada. Dicho desbalance provocó capacidades de predicción dispares entre las clases beneficiando aquellas de mayor frecuencia en la base.

El sistema creado fue desarrollado completamente a través del software Python apoyado en la librería Gensim para el desarrollo del modelo LDA y la librería Scikit-learn para los algoritmos de aprendizaje supervisado. Gracias a esta forma de desarrollo, la componente predictora del sistema queda preparada para predecir como para actualizarse en futuro con nuevos requerimientos al call center.

Para mi familia y don Luis

Agradecimientos

En primer lugar, quiero agradecer a mis profesores guías: Claudio Garretón, Marcos Orchard y Richard Weber, quienes siempre demostraron interés y preocupación por el desarrollo de este proyecto.

Agradezco a todo el equipo de Entel, quienes hicieron del último año una constante instancia de aprendizaje y nuevos desafíos, pero siempre de forma amigable, empática, preocupada y generosa.

Agradezco también a los profesores Patricio Acuña, Luis Reyes y Claudio Ortiz por el fuerte apoyo que me dieron cuando dudé respecto a mis capacidades. Mi gusto por las matemáticas y sus aplicaciones no existiría sin su participación en mi educación escolar.

Finalmente, agradezco a mi familia por siempre apoyarme tanto en mi educación como proyectos personales. Quien soy y el lugar donde estoy se deben netamente a la crianza que me otorgaron.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción

1.1. Motivación

Conforme el paso de los años y la rápida evolución de la tecnología, la cantidad de servicios asociados a la industria de las telecomunicaciones ha estado en constante aumento. Lo que alguna vez se limitó a la simple entrega del servicio de voz vía conmutación de circuitos hoy se ve compuesto por nuevas aristas tales como la maximización de cobertura, un buen servicio de datos móviles y servicios de valor agregado tales como las redes sociales entre otros.

Como consecuencia de estos cambios en la tecnología, los operadores de redes móviles han diversificado y mejorado constantemente sus servicios a modo de poder dar abasto a las nuevas necesidades de la red y los consumidores.

Complementario a la capacidad de entrega de un buen servicio se encuentra la capacidad de solución de problemas dentro del mismo. En pro de ambas capacidades, los operadores optan por los Call Center uno de sus principales canales de interacción y resolución de problemas.

En dicho contexto, el operador móvil Entel requiere optimizar la cadena operacional involucrada en la solución de los requerimientos realizados al Call Center, el cual pese a ser funcional en términos simples, no explota el total potencial de la información recibida y se estanca en cuanto a una falta de estandarización de la información entregada por él mismo, siendo este un problema transversal de la industria.

En el caso particular de este trabajo de título, el problema abordado, es la creación de un algoritmo computacional que automatice la interpretación de "pseudo-transcripciones"de conversaciones entre los clientes y los operadores del Call Center de Entel en pos de generar nueva información de entrada para un motor capaz de modelar la relación entre los motivos de contacto de los clientes (entregados por el sistema desarrollado) y los problemas de servicio reales que estos experimentan.

1.2. Alcance

El presente trabajo se basa en la utilización de los requerimientos de clientes post pago y prepago durante el año 2018 al Call Center de Entel para la creación de un programa capaz de clasificar dichas interacciones de forma automática.

Clasificar estas interacciones es de interés para la empresa puesto que permite analizar la relación entre los motivos de contacto de los clientes, los problemas de servicio reales que estos experimentan y a la calidad del servicio recibido.

Este trabajo pretende ser la base y documentación de la creación de una herramienta capaz de traducir información no estructurada (pseudo-transcripciones de llamadas) a entradas de interés al momento de modelar la métrica más importante para la empresa en lo que respecta a su servicio, la satisfacción de los clientes.

La construcción del modelo se lleva a cabo totalmente de forma computacional trabajando con Python y R. Por ser un problema de minería de textos, se trabaja con computadores en la nube, lo cual permite una mayor cantidad de recursos computacionales y facilita la implementación de los códigos correspondientes.

Pese a ser por esencia un problema de aprendizaje no supervisado, se procede a rotular una cantidad cercana a 10.000 requerimientos realizados en las categorías de interés para la empresa, lo cual entrega nuevas formas de cuantificar el rendimiento del algoritmo creado.

1.3. Objetivos

1.3.1. Objetivos generales

Creación, optimización y puesta en producción de un sistema capaz de clasificar las interacciones que tienen los clientes con el Call Center de Entel según la naturaleza de sus llamadas, de forma automática.

Automatización del proceso de entrenamiento de dicha herramienta para permitir actualizaciones y vigencia de la misma a través del tiempo.

1.3.2. Objetivos específicos

Elaboración de un sistema capaz de modelar tópicos, capaz de diferenciar los principales motivos por los que una persona llama al Call Center, siendo estos a la fecha: problemas de facturación y cobros, problemas del servicio de voz, problemas del servicio de datos y otros requerimientos relacionados a servicio de valor agregado y canales.

Optimización del modelo elegido en todas las aristas de elección que a él respectan. Apuntando del algoritmo definido de forma tal que su desempeño pueda maximizarse en cuanto a las capacidades demandadas por la empresa.

Automatización de todos los procesos relacionados a recolección, preprocesamiento y clasificación de los requerimientos de forma tal que esta herramienta quede en estado de producción para la empresa.

1.4. Resultados esperados

Los resultados esperados para este trabajo de título son:

- Generar un sistema capaz de clasificar de forma automática el motivo de llamada de un cliente al call center de Entel en función de la naturaleza del llamado es decir, como reclamo por servicio de voz, datos, cobros y facturaciones, canales o solicitud de algún requerimiento.
- Llevar el sistema a un entorno productivo, facilitando un posterior entrenamiento periódico del mismo en función de los nuevos requerimientos que lleguen al call center.
- Permitir auditar la calidad de la operación del call center de Entel en cuanto a la capacidad que este posee de caracterizar el real motivo de llamada de los clientes.
- Lograr establecer indicadores mensuales respecto a los principales motivos de llamada por parte de los clientes al call center de la compañía.

1.5. Estructura de la memoria

El presente informe se divide en 4 capítulos y se estructura de la siguiente manera.

En el Capítulo 1 se detalla la motivación para realizar el proyecto y se establecen los alcances, objetivos y resultados esperados.

El Capítulo 2 entrega los antecedentes relativos al análisis de tópicos, sus objetivos y como se desarrolla actualmente. Además de ello se presenta marco teórico técnicas implementadas.

El Capítulo 3 explica y muestra la creación de los diversos modelos creados para resolver el problema, compara los resultados obtenidos para cada con métricas de desempeño respectivas. Posteriormente define cual es el modelo elegido y los resultados obtenidos en su puesta en producción.

Finalmente, en el capítulo 4 se presentan las conclusiones relacionadas a la totalidad del trabajo y se proponen trabajos futuros a realizar.

Capítulo 2

Antecedentes relativos al análisis de tópicos

Las llamadas al call center se producen por clientes que desean hacer un requerimiento, o reclamar por una deficiencia del servicio entregado por la compañía. Como consecuencia, se trabaja con datos de connotación principalmente negativa o neutra por lo que se descartan técnicas de minería de datos orientadas al análisis de sentimientos, aquí el objetivo es discriminar los motivos vía breves transcripciones por lo que se plantea un problema de análisis de tópicos.

El análisis y modelamiento de tópicos es un área de investigación y aplicación de gran relevancia dentro de la minería de textos y datos. Dicha técnica permite dar provecho a información no estructurada, al día de hoy el tipo de dato más masivo disponible, fuentes de información no estructurada son cualquier tipo de instancia de reseñas o comentarios en sí generados por personas, tales como las encuestas, redes sociales o la atención al cliente de cualquier servicio.

2.1. Información involucrada en problemas de procesamiento de lenguaje natural

La información utilizada en los problemas de tipo *Natural Language Processing* (a partir de ahora referidos como NLP) se caracteriza principalmente por ser discreta, no existe continuidad en oraciones, palabras o letras que componen los documentos.

Las formas más comunes de representación de las mismas son: indicadores que toman el valor de 0 o 1 según el cumplimiento de alguna condición (e.g la aparición de la palabra en cada documento); vectores, que resultan de un mapeo desde el vocabulario original a un espacio multidimensional, permitiendo establecer relaciones entre palabras (e.g "hombre es a rey", como "mujer es a reina").

2.2. Estado del arte análisis de tópicos

Dentro de las técnicas utilizadas para llevar a cabo el análisis de tópicos, *Latent Dirichlet Allocation* (a partir de ahora referida como LDA) representa la más simple y popular [1]. LDA consiste en un modelo bayesiano jerárquico de tres niveles el cual establece que cada muestra (documento) se compone por una mezcla de los tópicos y cada tópico se representa por una distribución de probabilidades para cada palabra del diccionario.[2]

Respecto a las herramientas utilizadas hoy en día para abordar este problema se identifica el uso de LDA en un escenario similar correspondiente a la asignación de temas de conversación para transcripciones de diálogos de conversaciones telefónicas [3]. En esta implementación se procedió a etiquetar manualmente las conversaciones bajo los tópicos de interés de forma tal que este modelo no supervisado pudiera evaluarse como uno supervisado una vez asignados nombres a los tópicos que este asignaba a las muestras.

El resultado obtenido para dicha implementación fue un *accuracy* ponderado de 83,01 % sobre 4 tópicos en un total de 83 conversaciones telefónicas como conjunto de prueba, habiendo utilizado 125 conversaciones como conjunto de entrenamiento.

Inicialmente para el método LDA, la inferencia bayesiana se realizaba vía un algoritmo variacional bayesiano creado por David Blei [2], sin embargo, se discuten como posibles mejoras a implementar los acercamientos a la distribución posterior mediante técnicas de muestreo (sampling) tal como el algoritmo de muestreo Gibbs (MCMC) por Griffiths y Steyvers [4], este nuevo método requiere de más cálculos computacionales (ergo, más lento), sin embargo, demostró ser capaz de llegar a resultados menos sesgados y más precisos, dichas capacidades junto a la vigencia del método LDA motivaron durante el año 2014 a la creación de nuevas interfaces para la implementación del mismo.

En la actualidad, LDA junto a otros modelos bayesianos derivados de él siguen siendo implementados para atacar los problemas de análisis de tópicos, una excepción a tomar en cuenta es el modelo *Additive Regularization of Topic Models* (ARTM), modelo no bayesiano basado en una regularización de la estimación de la máxima verosimilitud y probado sobre breves textos [5], este

modelo se puede implementar vía la librería de software libre BigARTM.

Pese a las opciones que ofrece BigARTM para implementar el método ARTM se procede con LDA debido a la existencia de publicaciones que verifican el uso de dicha técnica para problemas afines al objetivo de este trabajo de título.

En cuanto a las herramientas a utilizar para implementar LDA, se opta por trabajar en el lenguaje de programación Python apoyado en las librerías *Natural Language Tool Kit* (NLTK) [6] y Gensim [7], ambas ampliamente utilizadas, con gran documentación y poseedoras de una constante fuente de apoyo en *stackoverflow* conocida comunidad de desarrolladores informáticos.

Aun cuando existen diversas técnicas para enfrentar los problemas de minería de textos, el preprocesamiento de la información es vital en todos ellos, parte de las actividades que se incluyen en el preprocesamiento de textos incluyen:

- Inclusión de N-gramas, generados en función de las probabilidades condicionales de la aparición consecutiva de las palabras en la colección de documentos.
- Creación de un corrector ortográfico para el lenguaje español el cual cumpla con su labor de corrección en tiempos considerables como óptimos.
- Exploración de las stopwords atingentes al contexto del problema planteado, debiendo realizarse esta actividad de forma manual y empírica.
- Creación de una lista de palabras denominadas como *lista blanca* la cual considerará palabras o siglas que no existen en el lenguaje Español pero son de sumo interés para este problema (e.g *spotify*, *whatsapp*, o servicios de valor agregado abreviados como VAS).
- Implementación de técnicas de lematización y stemming, novedosas para el lenguaje español.

Se motiva además la realización de un buen motor de preprocesamiento de textos para su posterior uso en otras instancias requeridas por Entel.

2.3. Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation consiste en una técnica para llevar a cabo análisis de tópicos dentro de textos, esta fue creada el año 2003 por los investigadores del área Andrew Ng, Michael Jordan y David Blei. Todo el desarrollo a continuación está basado directamente en el paper original *Latent Dirichlet Allocation* [2]

La esencia de este modelo consiste en decir que, para un grupo de textos, cada uno de ellos puede ser modelado por una distribución de tópicos y cada uno de estos tópicos puede ser modelado por una distribución de palabras. Dichos tópicos y sus distribuciones son variables latentes del total de datos, lo que da origen al nombre del modelo.

Se presenta la notación a utilizar a lo largo del presente documento, notación que fue creada por los autores del modelo probabilístico generativo.

- Documento (D): Observación o muestra de carácter textual (e.g comentarios, reseñas, correos

o requerimientos para este caso).

- Corpus (C): Colección de todos los documentos a trabajar.
- Vocabulario (V): Todas las palabras únicas encontradas en el corpus posterior al procesamiento.
- Matriz término documento: Matriz cuyas filas son documentos y cuyas columnas son cada palabra del vocabulario.

Sean k-tópicos, $B_{1:k}$ son distribuciones de probabilidad sobre un vocabulario fijo, dibujadas por una Dirichlet (η). El modelo asume que cada documento $D \in C$ es generado por el siguiente proceso generativo:

1. Escoger mezcla de tópicos θ^{d} de una distribución sobre un (K) simplex tal como una Dirichlet (α).
2. Cada una de las palabras del documento se genera escogiendo una asignación de tópico z desde una distribución $Multinomial(\theta^{\mathrm{d}})$ y posteriormente una palabra w desde una $Multinomial(\beta_z)$

Gráficamente este modelo se puede representar según la siguiente estructura:

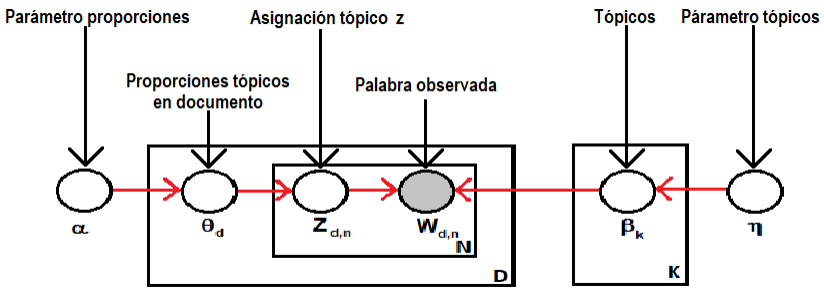

Figura 2.1: Esquema gráfico LDA, modelo jerárquico bayesiano.

De esta forma la distribución conjunta de una mezcla de tópicos θ junto a un set de N tópicos z y un set de N palabras w se define por:

$$p(\theta, z, w|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta) \tag{2.1}$$

Integrando la ecuación (2.1) sobre θ y sumando sobre z se obtiene la distribución marginal del documento como:

$$p(w|\alpha, \beta) = \int p(\theta|\alpha) \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n|\theta) p(w_n|z_n, \beta) \right) \mathrm{d}\theta \tag{2.2}$$

Consecuentemente, se multiplican las probabilidades marginales de cada uno de los documentos para obtener la probabilidad del corpus (C):

$$P(C|\alpha, \beta) = \prod_{\mathrm{d}=1}^{M} \int p(\theta|\alpha) \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n|\theta) p(w_n|z_n, \beta) \right) \mathrm{d}\theta_{\mathrm{d}} \tag{2.3}$$

Finalmente la inferencia en LDA se realiza mediante el computo de la distribución posterior de las variables ocultas dado un documento:

$$p(\theta, z|w, \alpha, \beta) = \frac{p(\theta, z, w|\alpha, \beta)}{p(w|\alpha, \beta)} \tag{2.4}$$

Debido a que dicha ecuación no puede ser computada en dicha forma, se propone utilizar inferencia varacional. La idea tras ello es buscar cotas inferiores variables para la log-likelihood que sean tan cercanas a ella como sea posible. Finalmente dicho problema se resuelve mediante la minimización de la divergencia de Kullback-Leibler (KL) entre la distribución varacional y la distribución a posteriori original.

2.4. Online Latent Dirichlet Allocation (OLDA)

Online Latent Dirichlet Allocation corresponde a una técnica creada a partir de LDA en donde el proceso de aprendizaje ha sido modificado pasando del esquema clásico de resolución vía un esquema variacional bayesiano a un esquema nuevo basado en optimización estocástica mediante un gradiente.

El gatillante para la creación de esta nueva técnica de optimización corresponde a las dificultades que poseen los esquemas variacionales bayesianos al momento de enfrentar bases de datos grandes. Esto como consecuencia de que su algoritmo de inferencia requiere un paso completo por la totalidad del corpus antes de poder actualizar los parámetros del modelo.

Tal como plantean los autores en [8], se replantea el algoritmo que da solución al problema de estimar la distribución posterior minimizando la distancia KL entre la aproximación q(z,θ,β) de la distribución posterior y la distribución posterior real p(z,θ,β|w,α,η).

El algoritmo anteriormente utilizado por la técnica variacional se muestra en el algoritmo 1.

Algoritmo 1 Batch variational Bayes para LDA

Inicializar λ aleatoriamente.
mientras mejora relativa en $\mathcal{L}(w, \phi, \gamma, \lambda) > 0{,}00001$ **ejecutar**
  Paso E:
  para d = 1 para D **ejecutar**
    Inicializar $\gamma_{\mathrm{d}k} = 1$ (constante 1 arbitraria.)
    repetir
      Definir $\phi_{\mathrm{d}wk} \propto \exp\{\mathbb{E}_q[\log\theta_{\mathrm{d}k}] + \mathbb{E}_q[\log\beta_{kw}]\}$
      Definir $\gamma_{\mathrm{d}k} = \alpha + \sum_w \phi_{\mathrm{d}wk} n_{\mathrm{d}w}$
    hasta $\frac{1}{K}\sum_k |\text{cambio en}\gamma_{\mathrm{d}k}| < 0{,}00001$
  fin para
  Paso M:
  Definir $\lambda_{kw} = \eta + \sum_{\mathrm{d}} n_{\mathrm{d}w}\phi_{\mathrm{d}wk}$
fin mientras

La nueva técnica implementada para la implementación de OLDA se presenta en el algoritmo 2.

Algoritmo 2 Online variational Bayes para LDA

Definir $\rho_t \triangleq (\tau_0 + t)^{-k}$
Inicializar λ aleatoriamente.
para $t = 0$ a ∞ **ejecutar**
  Paso E:
  Inicializar $\gamma_{tk} = 1$ (constante 1 arbitraria.)
  repetir
    Definir $\phi_{twk} \propto \exp\{\mathbb{E}_q[\log\theta_{\mathrm{d}k}] + \mathbb{E}_q[\log\beta_{kw}]\}$
    Definir $\gamma_{tk} = \alpha + \sum_w \phi_{twk} n_{tw}$
  hasta $\frac{1}{K}\sum_k |\text{cambio en}\gamma_{tk}| < 0{,}00001$
  Paso M:
  Computar $\tilde{\lambda}_{kw} = \eta + D\mathrm{d}_{tw}\phi_{twk}$
  Definir $\lambda_{kw} = (1 - \rho_t)\lambda + \rho_t\tilde{\lambda}$
fin para

En este caso, al recibirse el vector número t n_t de palabras en su formato **Bag of Words** (BOW) se realiza el paso E para encontrar óptimos locales para los valores γ_t y ϕ_t aún manteniendo λ fijo. Se computa $\tilde{\lambda}$ y se recomputa el valor de λ ponderándolo con el nuevo $\tilde{\lambda}$ (primera linea), dicha ponderación se regula con el parámetro $k \in (0{,}5, 1]$ el cual actúa como tasa de olvido de muestras anteriores.

2.5. LDAvis

LDA tiene la particularidad de ser un modelo jerárquico bayesiano de tres niveles, gracias a ello este modelo permite asociar más de un tópico a cada uno de los documentos.

Pese a lo anterior y para facilitar la comprensión de los resultados generados por el modelo, se creó la herramienta *LDAvis* la cual permite visualizar los tópicos generados en un espacio dos-dimensional con formas de circunferencia mientras que indica las palabras más sobresalientes para cada tópico según métricas definidas.

Al crear e implementar el modelo uno debe definir la cantidad de tópicos latentes que desea encontrar, sin embargo, de que tratan y en que consisten cada uno es algo invisible al operador hasta que se implementan técnicas como esta para examinar empíricamente que palabras los componen y así bautizarlos con alguna etiqueta formal.

En este espacio el área de dichas circunferencias representa la cantidad de documentos D los cuales tienen mayor probabilidad de haber sido generado por esos tópicos únicamente. Además, se permite visualizar las palabras más sobresalientes para cada uno de los tópicos ordenadas según dos métricas, **saliency** y **relevance**.

En primer lugar, el saliency de las palabras presentes en un tópico fue definido en 2012 por Chuang, Manning y Heer[9] en función de la probabilidad de aparición de la palabra multiplicada por su *distintivity*, siendo esta última a divergencia de Kullback-Leibler (KL) entre P(T|w) y P(T) donde T se refiere a un tópico latente y w a una palabra o término específico. En consecuencia se define el saliency de una palabra como:

$$saliency(w) = P(w)x \sum_{T} P(T|w) log \frac{P(T|w)}{P(T)} \tag{2.5}$$

Por su parte, dado un valor $\lambda \in [0, 1]$ el relevance de un término se define por Carson Sievert y Kenneth Shirley en [10] como:

$$r(w, k|\lambda) = \lambda(\phi_{kw}) + (1 - \lambda) log \left(\frac{\phi_{kw}}{p_w} \right) \tag{2.6}$$

En (2.6), λ determina el peso que se le entrega a la probabilidad de una palabra w bajo un tópico k con respecto a su **lift**. Mientras que lambdas cercanos a 1 retornan un ranking de términos caracterizado por su probabilidad específica para dicho tópico, valores de lambda más cercanos a 0 retornan un ranking de términos enfocado en el lift de los mismos.

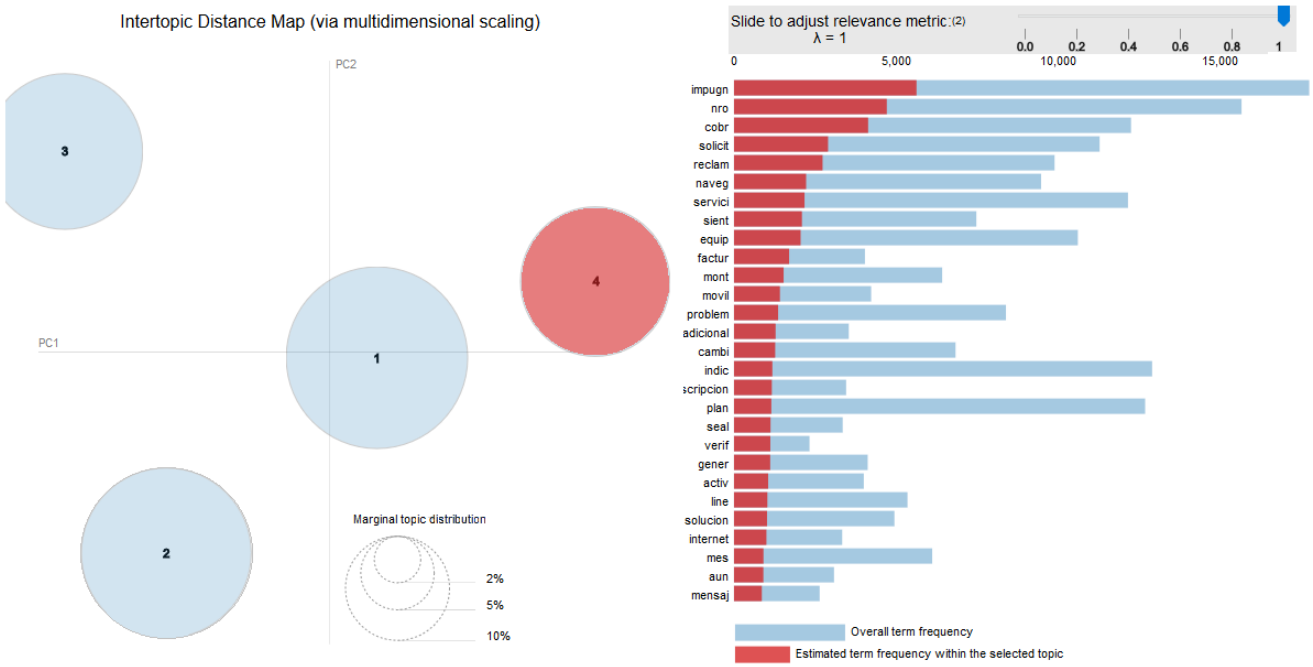

Figura 2.2: Visualización herramienta LDAvis.

2.6. Métricas de desempeño

El estudio de desempeño de los diversos modelos probados se basó en un abanico de métricas, el carácter de estas se puede dividir en dos ramas.

Las métricas utilizadas en aprendizaje supervisado y aquellas que son intrínsecas del método LDA.

2.6.1. Entrenamiento supervisado

Debido a que el modelo LDA simplemente retorna el vector de composición (mezcla) de los tópicos sobre un documento se decide tomar aquellos tópicos con mayor probabilidad para hacer la elección de clase.

Consecuentemente, se consigue una etiqueta la cual rotula el documento mediante LDA sin embargo aún debe haber una etiqueta real"sobre los mismos documentos.

Para compensar esta ausencia y de esa forma poder utilizar las métricas tradicionalmente usadas en problemas supervisados corresponde entonces etiquetar manualmente la clase real a la cual pertenece cada uno de los documentos. Una vez echo eso se permite utilizar las siguientes métricas.

- *Accuracy*: Esta métrica indica la proporción entre las predicciones correctamente realizadas

y el total de predicciones realizadas, de esta forma el accuracy se define como:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.7}$$

- *Precision*: Para una clase elegida, se calcula el *precision* como la cantidad de muestras correctamente etiquetadas como positivas del total de muestras clasificadas como positivas. Es decir:

$$Precision = \frac{TP}{TP + FP} \tag{2.8}$$

  Por su definición, al optimizarse esta métrica se está disminuyendo el total de falsos positivos.
- *Recall*: Esta métrica cuantifica cuantas muestras de una clase específica fueron catalogadas como dicha clase con respecto al total de muestras originalmente de dicha clase. Es decir:

$$Recall = \frac{TP}{TP + FN} \tag{2.9}$$

  Al contrario del *precision*, esta métrica da énfasis en disminuir los falsos negativos de la clase estudiada.
- *Fscore*: Corresponde a una métrica originada a partir del recall y la precision.

$$Fscore = 2\left(\frac{precision \cdot recall}{precision + recall}\right) \tag{2.10}$$

  Así, esta métrica depende tanto de los falsos positivos como falsos negativos para aumentar en valor. Consecuentemente, predictores con un valor alto de f-score serán el modelo ideal la mayoría de las veces.
- Top2v2: Las métricas anteriores se usan cuando la variable objetivo es solo una. La métrica Top2v2 se crea para cuando las variables objetivo son dos (e.g dos posibles reclamos a call center en una sola llamada).

$$\text{Top2v2} = \frac{\sum_{\text{i}=1}^{I} \delta(\text{muestra}_\text{i}) + \sum_{j=1}^{J} \zeta(muestra_j)}{2 \cdot (I + J)} \tag{2.11}$$

  En la anterior, i=[1,2,3...I] representa las muestras que poseen solo una etiqueta original mientras que j=[1,2,3,...J] representa aquellas muestras con dos etiquetas originales.

  Además, se definen las funciones a trozos $\delta(\text{muestra}_x)$ y $\zeta(\text{muestra}_x)$ según

$$\delta(\text{muestra}_x) = \begin{cases} 0 & \text{si no es correcta la clasificación} \\ 1 & \text{si clasificación es correcta} \end{cases}$$

$$\zeta(\text{muestra}_x) = \begin{cases} 0 & \text{si no hay clasificaciones correctas} \\ 1 & \text{si solo 1 clasificación es correcta} \\ 2 & \text{si dos clasificaciones son correctas} \end{cases}$$

  De esta forma, la métrica Top2v2 actúa como un *accuracy* el cual compara dos pares de columnas, un par de etiquetas reales y un par de clasificaciones respectivas. Esta métrica no discrimina ni atribuye importancia al orden en que se mencionan las clases de interés, solo se preocupa de la mención.

- Desempeño especifico: También de creación particular, esta métrica calcula la cantidad de aciertos para cada una de las posibles clases con respecto a todas las predicciones realizadas entre todas las muestras.

  En consecuencia, esta métrica retorna para cada clase un recall calculado sobre todas las posibles columnas de predicción y etiqueta real.

$$\text{Desempeño especifico} = \frac{TP_{1-2-3}}{TP_{1-2-3} + FN_{1-2-3}} \tag{2.12}$$

  Esta métrica carece de sentido en un escenario donde el modelo predictor retorna todas las clases como etiqueta. Para evitar dicho comportamiento se eliminan las predicciones que el modelo realiza bajo un umbral de probabilidad de pertenencia a la clase por etiquetar.

  Por ejemplo, asumiendo que para un problema donde la variable objetivo toma tres valores, el utilizar una probabilidad de pertenencia mínima para predicción igual a 0,34 permitirá de que el algoritmo asigne un máximo de 2 clases a las muestras sobre las que prediga.

2.6.2. Entrenamiento no supervisado

Dentro de las métricas que se pueden implementar para evaluar el desempeño del algoritmo utilizado se encuentran:

- *Perplexity*: Correspondiente a la exponencial de la entropía, esta métrica puede verse como el inverso de la verosimilitud por palabras [2]. Definiéndose como:

$$perplexity(C_{test}) = exp(-\sum_{m} \log p(w_m)/(\sum_{m} |w_m|)) \tag{2.13}$$

  De esta forma, valores menores de perplexity serán mejores, sin embargo, esta métrica sirve en la medida que compara modelos sobre un mismo problema, es decir, es relativa.
- *U-mass*: Métrica que cuantifica coherencia del modelo basándose en las palabras y su presencia para cada uno de los tópicos modelos. Así, se define la coherencia *Umass* como [11].

$$C_{\text{UMass}} = \frac{2}{N \cdot (N-1)} \sum_{\text{i}=2}^{N} \sum_{j=1}^{\text{i}-1} \log \frac{P(w_{\text{i}}, w_j) + \varepsilon}{P(w_j)} \tag{2.14}$$

  De esta forma, esta métrica será interesante de utilizar en la medida que hablará sobre la coherencia de cada tópico conforme a las palabras que lo componen. Se destaca además que esta métrica tendrá valores negativos los cuales reflejarán un mejor desempeño conforme sus valores se acerquen a 0.

2.7. Técnicas de preprocesamiento de texto

Sin importar la herramienta especifica utilizada, todas las tareas que involucran procesamiento y trabajo sobre lenguaje natural requieren de un correcto preprocesamiento para su correcto funcionamiento. La idea esencial de este es el poder disminuir la dimensionalidad de las entradas a los modelos mediante la reducción del vocabulario (V) con el que se trabaja.

A continuación se presenta las principales herramientas utilizadas en pos de lograr una disminución de la dimensionalidad del vocabulario mientras se trata de resguardar la riqueza de los datos en la medida de lo posible:

2.7.1. Corrección ortográfica

Esta herramienta es de vital importancia en contextos donde los documentos puedan presentar diversas faltas de ortografía, en la medida que la gente se equivoca en la forma de escribir una palabra se aumentará de forma innecesaria la dimensionalidad del vocabulario global.

Debido a su mayor facilidad de implementación, al día de hoy múltiples esquemas de corrección ortográfica optan por una la corrección ortográfica a nivel de palabras individuales, el funcionamiento de estos consiste en el uso de una gran tabla de búsqueda con palabras del lenguaje, tomándose una palabra se revisa la presencia de ella en dicha tabla y en caso de respuesta negativa se procede a realizar transformaciones sobre esta buscando modificarla a alguna palabra que si se encuentre en dicha tabla.

Dentro de estos correctores se decide utilizar el primer corrector ortográfico creado por Google bajo la dirección de Peter Norvig [12]. El funcionamiento de este se explica básicamente en 4 partes:

1. Revisión de pertenencia palabra a lista de palabras válidas para el lenguaje objetivo.
2. En caso de negativa en paso 1, realización de las siguientes transformaciones sobre dicha palabra:
   - Divisiones entre cada carácter, generando un par de palabras donde se opta por validar la más larga generada, creando palabras como se ve en tabla 2.1

allár	a - llár	al - lár	all - ár	allá - r

Tabla 2.1: Divisiones generadas sobre una palabra.

   - Eliminaciones de caracteres, probando por casilla se probará eliminar caracteres de la palabra como se ve en la Tabla 2.2

árbols	rbols	árols	árbls	árbos	árbol

Tabla 2.2: Eliminaciones generada sobre una palabra

   - Transposiciones sobre los caracteres que componen la palabra tal como se muestra en la Tabla 2.3.

zaf	az	faz	zfa

Tabla 2.3: Aplicación de transposiciones a palabra genérica.

   - Reemplazos sobre los caracteres que componen la palabra como en la Tabla 2.4.

kasa	aasa	basa	casa	...	zasa	kbsa	kcsa	...

Tabla 2.4: Aplicación permutaciones corrector ortográfico

- Adición de caracteres en todas las posiciones de la palabra, ejemplo en Tabla 2.5.

kasa	akasa	kaasa	kaasa	kasaa	kasaa	...

Tabla 2.5: Ejemplo de aplicación adición caracteres en posiciones validas palabra, se insertan todas las posibles letras en cada espacio entre caracteres.

3. Finalizados todos los pasos de 2. sobre una palabra que se desea corregir, se genera una lista con las repeticiones únicas de las palabras resultantes y se realiza nuevamente el paso 2. sobre cada una de las palabras que componen esta.
4. Se toma nuevamente una lista que incluya las palabras únicas obtenidas tras los pasos 2. y 3. y se elige como corrección correcta aquella que resuelva el siguiente problema de optimización.

$$\text{argmax}_{c \in \text{candidatos}} \, \text{P}(c|w) \tag{2.15}$$

En este, se plantea elegir aquella corrección c que tenga la mayor probabilidad de ser la corrección correcta, por Bayes este problema es similar a:

$$\text{argmax}_{c \in \text{candidatos}} \, \frac{\text{P}(w|c) \, \text{P}(c)}{\text{P}(w)} \tag{2.16}$$

En 2.16 la componente P(w|c) representa el modelo de error en la corrección i.e que el autor haya querido decir la palabra con la que se corrige por ejemplo, corregir ïente.^{a} "siente.^{es} más probable que ir de ïente.^{a} çliente"puesto que la primera corrección solo agrega un carácter mientras que la segunda agrega dos.

El corrector ortográfico implementado por Norvig utiliza dicha convención para diferenciar el P(w|c) de distintos candidatos a corregir.

Por su parte P(w) representa la probabilidad de la palabra original, al ser igual para cada candidato es un factor que se ignora.

P(C) cuantifica la probabilidad de la palabra correctora candidata C dentro del idioma en cuestión, mientras que corregir ïente"por çliente"puede tener mayor sentido en un contexto de call center, en el idioma español la palabra siente es más utilizada que cliente por lo tanto su P(c) será mayor.

La razón por la que decide utilizar este tipo de corrección ortográfica la cual ignora la semántica dentro de las oraciones se debe al nivel de dificultad que conlleva dicha opción.

Para una revisión a mayor detalle del funcionamiento de un corrector ortográfico de uso productivo se puede revisar [13].

2.7.2. Remoción stopwords

Independientemente de la naturaleza del problema a enfrentar, en el español al igual que en otros idiomas existen palabras conectoras y con funcionalidades lingüísticas ilativas para efectos

de una oración.

Pese a que estas palabras entregan información necesaria para la correcta lectura por parte de una persona, ellas sobran para efectos de un algoritmo de conteo de palabras puesto que no le entregan información.

Se decide en consecuencia generar listas de palabras comúnmente denominadas *stopwords* la cual contendrá aquellas palabras las cuales no entregan información al modelo y por ende serán removidas de los distintos documentos al ser identificadas.

2.7.3. Lematización

Otra técnica clásica para reducción de dimensionalidad de un vocabulario corresponde a la lematización.

Esta técnica busca reducir el vocabulario existente mediante la acción de llevar el lenguaje a su forma más sencilla, de forma genérica se pueden mencionar las siguientes acciones principales:

- Desconjugación verbal: Llevando todos los verbos a sus formas infinitivas.
- Remoción de sufijos que derivan en cantidades de algo.

De esta forma, la lematización generará que hasta 6 palabras se tornen una como se ve en la Tabla 2.6.

Palabra original	corren	corrieron	correrán	corriendo	corre
Palabra lematizada	correr	correr	correr	correr	correr

Tabla 2.6: Ejemplo de aplicación lematización.

2.7.4. Stemming

La aplicación y uso de un stemmer es denominada en inglés como la acción de realizar *stemming*, al igual que la lematización esta busca reducir el vocabulario pero esta vez mediante la simple aplicación de reglas de estructura sobre un conjunto de letras unidas en una palabra.

Esta técnica fue diseñada originalmente para el idioma ingles debido a la simpleza del mismo, sin embargo, adaptaciones de la misma se han realizado para el lenguaje español. En la Tabla 2.7 se visualiza un uso de la misma.

Palabra original	cañon	cañones	camionero	camionera
Palabra stemmizada	cañon	cañon	camioner	camioner

Tabla 2.7: Ejemplo de aplicación stemming.

Aun cuando el idioma en que se originaron los stemmers es el inglés, se han creado mediante el lenguaje de programación **Snowball** diversas versiones alternativas de stemmers para distintos idiomas diversos idiomas cada uno con sus reglas específicas.

Para el caso del español, el proceder del stemmer se basa en tres principales pasos[14]:

1. Remoción estándar de sufijos.
2. Remoción sufijos especiales, donde se identifican aquellos de verbos que comienzan con la letra 'y' y además otros sufijos especiales como 'en', 'es', 'éis' y 'emos'.
3. Remoción de sufijo residual, donde se busca eliminar los siguientes sufijos 'os', 'a', 'o', 'á', 'i', 'ó'.

2.7.5. N-gramas

Pese a que disminuir la dimensión del vocabulario es un aspecto crítico para un buen desempeño del modelo, puede ocurrir que palabras por separado tengan distinto significado a las mismas en conjunto, un ejemplo de ello puede verse en las oraciones:

- El cliente llama para reclamar por el servicio técnico.
- El cliente reclama pues su teléfono no llama.

En las anteriores, el sentido de la palabra llama es totalmente distinto y sin embargo, un esquema como LDA no será capaz de diferenciar que la palabra llama es de gran y distinta importancia en la segunda oración con respecto a la primera.

En consecuencia, surge los bigramas y trigramas, nuevas palabras que representan 2 palabras originalmente separadas unidas por algún carácter especial como un "_". La forma en que se generan estos es mediante la aplicación de reglas estadísticas para la aparición de los pares de palabras en diversos documentos ponderadas por la aparición de las palabras separadas en el corpus global.

De esta forma, se busca en primer lugar generar pares de bigramas de palabras las cuales se repitan una cantidad de veces mínima a lo largo del corpus. Posteriormente se les valora según un puntaje definido por Mikolov et al. en [15]:

$$score(w_i, w_j) = \frac{\text{count}(w_i\ w_j) - \delta}{\text{count}(w_i)\ \text{count}(w_j)} \tag{2.17}$$

En 2.17, el valor δ es un simple límite inferior para el puntaje exigido de forma tal que variando dicho parámetro se exigirá una mayor aparición conjunto de ambas palabras. Además, se resta importancia a aquellos bigramas formados por palabras cuya aparición es recurrente a lo largo de todo el corpus.

Aún cuando existe δ para exigir una cota mínima de apariciones mínimas de las palabras en conjunto, se procede a establecer una cota sobre el score final de cada bigrama antes de añadirlos a los documentos.

La aplicación de esta técnica para generar bigramas consiste en re-implementarla una vez ya se han obtenido los bigramas de forma tal que apariciones consecuentes de bigramas y otras palabras formaran los trigramas en caso de estos superar el score mínimo definido.

2.8. Algoritmos de aprendizaje supervisado de máquinas

Al contrario del esquema de aprendizaje no supervisado, un esquema de trabajo de aprendizaje supervisado corresponde al escenario en que la variable que se desea predecir sí posee un valor real previamente asignado.

Dentro de toda la familia de técnicas de aprendizaje supervisado se presentan las tres que serán exploradas para explorar posibles mejoras sobre el sistema creado a partir de LDA.

2.8.1. Random Forest

Este algoritmo consiste en un grupo de árboles de decisión [16], y surge como una medida para palear la baja capacidad de generalización de los mismos.

La idea tras el funcionamiento de este algoritmo consiste en que si bien un árbol no siempre podrá generalizar de forma correcta, la votación resultante de múltiples modelos poco correlacionados podrán aproximarse mejor a la solución.

Para lograr obtener árboles poco correlacionados se toman dos medidas:

1. Antes de entrenar un árbol se genera un conjunto de entrenamiento a partir de un muestreo con reposición sobre el conjunto de entrenamiento original.
2. Durante el entrenamiento de cada árbol de decisión y para cada nodo se toma un subconjunto aleatorio de las variables predictoras.

Consecuentemente, el escenario producido será el de múltiples árboles sobre ajustados a sus respectivos conjuntos de entrenamiento y variables predictoras disponibles, conformando un 'bosque' capaz de generalizar de mejor forma.

Capítulo 3

Desarrollo de sistema automático de clasificación de motivos de llamada cliente-call center

3.1. Metodología de trabajo LDA para clasificación

3.1.1. Elección modelo y bases de datos

Como caracterización del escenario de interés, se presenta una base de datos no estructurados donde sus documentos consisten de oraciones sencillas en su mayoría de connotación negativa pues reflejan mayoritariamente problemas en la entrega del servicio.

En consecuencia de lo anterior, se opta por implementar LDA (modelo basado en conteo de palabras carente de mayor análisis semántico) como clasificador sobre los requerimientos del call center.

El problema a resolver presenta una dualidad en el tipo de cliente, por una parte están los clientes suscritos al servicio de la empresa y como contraparte se tiene a los clientes que prepagan el servicio. Cada tipo de cliente presenta distintos tipos de problemas en relación a la entrega del servicio. Consecuentemente, se debera crear un modelo para cada categoría de cliente.

En cuanto a las bases de datos, para la base de clientes de suscripción se toma un subconjunto de 90.000 documentos del total de requerimientos, mientras que para la base de clientes de prepago se toma un subconjunto de 60.000 documentos del total.

En ambos casos se dividen los subconjuntos en dos grupos: un 85 % de requerimientos destinados a entrenamiento y un 15 % de datos de validación. El etiquetado de los conjuntos de validación es manualmente realizado por personal capacitado.

Para la evaluación de desempeño, se trabaja con un esquema de aprendizaje semi supervisado, en

él se combinan la componente de aprendizaje no supervisado del algoritmo LDA con la metodología de ajuste de parámetros en función del desempeño del modelo LDA en modalidad de clasificador contrastado a las etiquetas generadas manualmente generadas.

Para llevar a cabo la clasificación, el modelo elige como clase aquel elemento que posea mayor porcentaje de influencia en el vector de mezcla de tópicos (ecuación 3.1), dicho vector es específico a cada uno de los documentos.

$$\arg\max_c \mathrm{p}(c|w) = \arg\max_c \mathrm{p}(w|c)\mathrm{p}(c) \tag{3.1}$$

El modelo LDA retorna los tópicos que mejor ajustan el modelo probabilístico generativo al corpus en cuestión, consecuentemente, se utiliza la herramienta LDAvis para permitir asociar nombres lógicos a cada uno de los tópicos caracterizados por modelo.

3.1.2. Modelo LDA

La creación global del sistema de clasificación involucra diversas etapas las cuales van desde la recolección de los datos hasta la devolución de los mismos al sistema con las etiquetas del modelo incluidas.

En la Figura 3.1 se muestra el esquema de trabajo propuesto para el desarrollo y creación del modelo LDA, este esquema de trabajo comprende el núcleo del sistema puesto que detalla tanto el aprendizaje inicial del mismo como su posterior actualización (habrá una implementación de este esquema para clientes prepagos y una para suscritos).

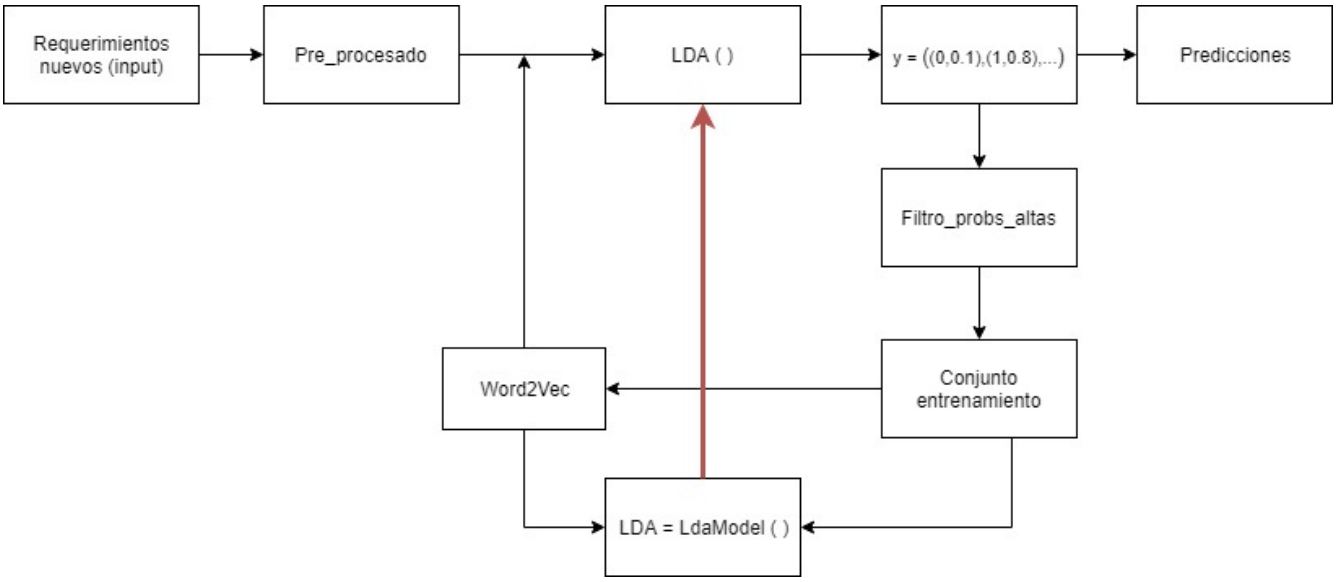

Figura 3.1: Esquema de trabajo modelo LDA para sistema de clasificación interacciones.

Disponiendo de las bases de datos, el primer paso a realizar es el preprocesado de los textos (detalle en capítulo 3.1.2.1 sobre preprocesamiento de los textos). Una vez preprocesados los docu-

mentos, se utilizan los conjuntos de entrenamiento respectivos para llevar a cabo el entrenamiento del modelo.

Para los clientes de prepago, se conoce a priori que predominan cuatro motivos de reclamos: servicios de voz, servicios de datos, canales y servicio técnico (alías otros) y bolsas y recargas. Por otra parte, para los clientes suscritos también se identifican cuatro motivos predominantes de reclamos: servicios de voz, servicios de datos, canales y servicio técnico y boletas y facturación.

Coloquialmente y para facilitar el orden en la presentación de resultados, se denominarán los motivos de reclamo en las clases:

- Voz, Datos, Otros y Facturación para clientes suscritos.
- Voz, Datos, Otros y Bolsas o Recargas para clientes de prepago.

Con los modelos entrenados, se traduce la salida de los mismos a una etiqueta (tópico) legible por el humano (detalle en capítulo 3.1.2.2).

Habiéndose completado los pasos anteriores se realizan las clasificaciones sobre los requerimientos de las bases de validación respectivas utilizando la ecuación (3.1). Utilizando dichas predicciones se generan las métricas de desempeño y se almacenan para futura comparación entre modelos.

Específicamente, la salida del sistema es un vector de tuplas en donde se indica cada una de las clases (motivos de llamado) junto a la probabilidad de que dicho documento se generase totalmente por cada uno, de esta forma, la predicción final son aquellas clases de mayor probabilidad dentro del vector mencionado que cumplan con tener al menos un 35 % de probabilidades de haber generado dicho documento.

Una vez se hayan entrenado y almacenado los desempeños en validación de suficientes modelos, se elegirá aquél con mayor Fscore ponderado como el modelo final. Dicho modelo final será actualizado sistemáticamente para palear los cambios que se producirán en la naturaleza de los reclamos al call center con el paso del tiempo.

Técnicamente, para realizar la actualización se implementará el modelo de *Online Latent Dirichlet Allocation* (OLDA) el cual tomará el modelo previamente entrenado y lo actualizará con nuevos requerimientos ponderando el nuevo aprendizaje con el que ya se tenía anteriormente.

En resumen, el ciclo completo de entrenamiento y uso del sistema comprenderá el entrenamiento del modelo tanto para clientes prepagos como suscritos, la ejecución de los sistemas sobre nuevos requerimientos y la actualización de ellos. En todas las etapas anteriores, se realizará el preprocesamiento debido de los requerimientos.

3.1.2.1. Preprocesamiento de los textos

Al momento de preprocesar los textos de los requerimientos se pueden implementar múltiples técnicas, las técnicas específicas utilizadas en este proyecto se presentan en el esquema 3.2.

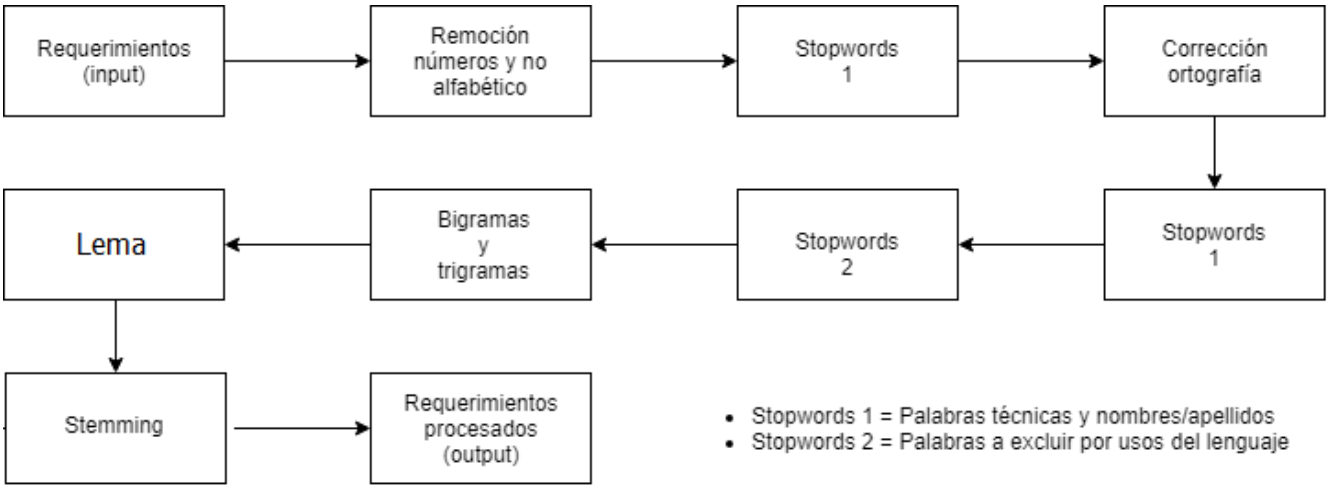

Figura 3.2: Esquema preprocesado textos de requerimientos clientes.

El preprocesamiento de los textos consiste en:

1. Remoción caracteres numéricos y especiales de los textos.
2. Remoción palabras vetadas lista stopwords1: Aquellas palabras de índole técnica que aportan a siguiente iteración atención cliente (códigos call center) pero no al sistema y también nombres de las personas que llaman, información también irrelevante.
3. Corrección ortográfica de los requerimientos, para ello se implementa el corrector diseñado por Peter Norvig.
4. Nuevamente se remueve palabras vetadas stopwords1 y posteriormente se remueven palabras lista stopwords2 (conectores del lenguaje español y derivados lingüísticos que no aportan información a oraciones).
5. Con el texto en su forma corregida y sin palabras de sobra, se procede a crear bigramas y trigramas (e.g sin_señal, celular_no_llama) mediante implementación propuesta por Thomas Mikolov.
6. Lematización de las palabras en oraciones, para ello, se utiliza un diccionario de todo el lenguaje español con su versión original y lematizada de las palabras.
7. Finalmente se aplica stemming sobre las palabras mediante el uso de un stemmer para el idioma español con reglas basadas en el stemmer de Martin Porter.

Tanto la técnica de stemming como la técnica de lematización son ampliamente utilizadas y recomendadas pues ayudan a disminuir la dimensionalidad del vocabulario, facilitándole la labor al modelo LDA.

3.1.2.2. Asignación automática de nombre a tópicos retornados

Para llevar a cabo la asignación de nombre a los tópicos retornados por cada uno de los modelos de forma automática se propone la siguiente metodología:

1. Crear diccionario de palabras características de cada tópico para cada escenario a estudiar.
2. Tras entrenar cada modelo, utilizar el diccionario correspondiente al escenario en cuestión para asignar una clase a cada uno de los tópicos.

Para crear dicho diccionario, iterar múltiples veces sobre un modelo genérico para cada uno de los escenarios propuestos para estudiar. Cada iteración consiste en: entrenar dicho modelo, retornar para cada uno de los tópicos resultantes sus palabras más probables dentro de las multinomiales respectivas (e.g Tabla 3.1).

Tópico	p1	p2	p3	p4
0	0,069 - 'client'	0,043 - 'movil'	0,042 - 'lin'	0,037 - 'si'
1	0,123 - 'bols'	0,105 - 'naveg'	0,046 - 'rrss'	0,040 - 'si'
2	0,067 - 'client'	0,043 - 'app'	0,042 - 'ingres'	0,042 - 'fech'
3	0,180 - 'bols'	0,140 - 'recarg'	0,029 - 'mont'	0,029 - 'compr'
4	0,082 - 'llam'	0,065 - 'problem'	0,054 - 'comun'	0,043 - 'region'
5	0,073 - 'client'	0,068 - 'sald'	0,053 - 'recarg'	0,051 - 'llam'

Tabla 3.1: Palabras más probables en distribuciones multinomiales asociadas a cada tópico.

Así, con la información de la Tabla 3.1 hay que evaluar que cada una de las filas se asemeje a un posible reclamo de alguna de las clases propuestas para las bases de suscritos o prepagos. En caso de semejanza proceder a utilizar LDAvis, caso contrario ajustar parámetros y reentrenar.

Con LDAvis se buscarán dos cualidades en cada modelo, en primer lugar la existencia de al menos 4 (o la cantidad establecida) grupos de esferas de tópicos en el mapeo de distancias inter-tópicos.

En segundo lugar, se ajustará el parámetro λ del modelo entre los valores 0,3 y 1 para estudiar la relevancia de diversas palabras en cada uno de los tópicos según la frecuencia total de apariciones que tienen en un tópico específico.

Dicho movimiento en la relevancia permitirá que palabras con baja frecuencia total en el corpus; pero con una gran fracción de su total de apariciones en uno solo de los tópicos puedan superar la importancia de palabras cuya probabilidad dentro de la multinomial es mayor principalmente como consecuencia de la gran cantidad de apariciones que tienen en el corpus.

Posteriormente, se tomará registro de las palabras más importantes con la herramienta para cada tópico y repetir el ejercicio con múltiples modelos. Finalmente, añadir aquellas palabras más repetidas en cada tópico al diccionario.

Una vez creado el diccionario, la asignación automática de tópicos se realizará en base al algoritmo 3:

Algoritmo 3 Asignación automática de nombres para tópicos generados por modelo LDA

Entrenar modelo LDA específico y generar tabla con palabras más probables en las multinomiales correspondientes.
para clase $_j$ en clases modelo LDA **ejecutar**
  para tópico $_{\text{i}}$ en diccionario tópicos **ejecutar**
    $p_{acumulada_{\text{tópico(i)}}} = \sum_{\text{i}}^{n} p_{palabra} \cdot \delta(\text{palabra}|\text{tópico}_{\text{i}})$
  fin para
  $\text{i}_{\text{máx}} = \text{i}$ para el i que cumple: p_acumulada_tópico(i) = máx(p_acumulada_tópico)
  nombre_clase $_j$ = tópico($\text{i}_{\text{máx}}$)
fin para

En el anterior, la función δ toma los valores de 0 o 1 según la palabra que se le entrega existe dentro del listado del tópico de estudio dentro del diccionario de tópicos. Por ejemplo, en el tópico de "facturación"del diccionario, se cumplirá que: δ(impuganción) = 1)

3.1.3. Escenarios a estudiar

Aún cuando la metodología para el preprocesamiento de textos fue detallada en el capítulo anterior (3.1.2.1), existen muchos parámetros de control sobre el modelo LDA a implementar por definir.

En pos de encontrar los parámetros óptimos para el diseño de los modelos LDA, se proponen diversos escenarios con características únicas tanto en los parámetros que los definen como en los preprocesamientos realizados para cada uno.

Se definieron inicialmente las características de la Tabla 3.2, estas se repetirán a lo largo de los cinco escenarios de estudio planteados para ambas bases de datos.

Rango frecuencias mínimas	N° de casos	Pasos por corpus	Prob. mínima
[50,...,1.650]	320	8	0,35

Tabla 3.2: Valores comunes a escenarios de estudio modelo LDA.

El rango de frecuencias mínimas contiene distintos valores mínimos de aparición de exigidos a las palabras en la totalidad del corpus de entrenamiento para ser tomadas en cuenta por el modelo.

El número de casos a estudiar consiste en la cantidad de modelos que se entrenaran variando la frecuencia mínima exigida siempre dentro del rango de frecuencias mínimas detallado, la división de dicho rango es en intervalos lineales por lo que cada caso consecutivo tomó una frecuencia mínima de 5 apariciones más que el anterior.

La cantidad de pasos por el corpus corresponde a las veces que el algoritmo recorrerá reasignando el tópico asociado a cada palabra dentro de cada uno de los documentos a una de las posibles clases o tópicos que generan dichos documentos, este parámetro puede entenderse como un concepto análogo a las épocas de entrenamiento en redes neuronales.

La probabilidad mínima corresponde al valor a partir del cual un valor de porcentaje de mezcla de una clase específica será tomado en cuenta para realizar una predicción.

Por ejemplo, para un valor de probabilidad mínima de $0,35$ en un escenario de cuatro tópicos donde se tiene un vector específico de salida del modelo de la forma [0,36 - 0,35 - 0,28 - 0,01], se clasificaran los tópicos asociados al elemento uno y dos del vector como motivo uno y dos de llamado respectivamente, mientras que los otros dos elementos no son tomados en cuenta.

La manipulación del valor de probabilidad mínima toma importancia puesto que conlleva dos consecuencias de interés, en primer lugar, disminuirla o aumentarla significará la aparición de más o menos segundas clasificaciones respectivamente.

Por otra parte, el disminuir o aumentar el valor en cuestión provocará primeras predicciones con menor o mayor certeza respectivamente, mayor exigencia puede significar mejor desempeño en la predicción, pero también el que no se realicen predicciones en algunas muestras.

Los escenarios a continuación representan el las diversas configuraciones de preprocesamiento de datos planteadas en pos de encontrar el mejor modelo LDA:

Escenario	Lema	Stemming	N-gramas	N° de tópicos
1	No	Si	No	4
2	Si	No	Si	4
3	Si	No	Si	5
4	Si	No	Si	6
5	Si	Si	Si	4

Tabla 3.3: Escenarios a estudiar variando técnicas de preprocesado aplicadas.

Para los 5 escenarios anteriores, el estudio de desempeño del algoritmo se enfoca principalmente en las métricas de Fscore y Top2v2. Además, se planea adjuntar los tiempos de entrenamiento relativos a cada escenario a modo de ponderar también las mejorías de desempeño con la factibilidad de uso de los modelos.

Se entrenan así un total de 1700 modelos LDA desde 0, los desempeños de cada uno de ellos se comparará con sus pares en los respectivos escenarios de preprocesamiento buscando elegir aquel que desempeñe mejor. Se destaca nuevamente que para cada uno de dichos casos, se preprocesa tanto el conjunto de entrenamiento como el de validación.

Finalmente se optará por preservar y actualizar a futuro aquel modelo que presente mejor desempeño de entre todos los modelos involucrados en los primeros 5 escenarios propuestos.

Para determinar el valor óptimo de cota inferior y score solicitado a los bigramas y trigramas para su integración se trabaja de forma manual estudiando las variaciones producidas sobre los corpus de estudio una vez que estos hayan tenido corregida su ortografía y removidas sus stopwords.

De forma independiente a los escenarios planteados anteriormente, se propone explorar la variación en el desempeño de los modelos LDA en función de la cantidad de pasos totales a través del corpus durante el entrenamiento de los mismos, para ello, se estudian los siguientes dos escenarios

exclusivamente en la base de clientes de postpago:

Escenario	Frecuencia Mínima	Rango de pasos	Lema	Stemming	N-gramas
6	786	[1,50]	Si	No	Si
7	666	[1,50]	No	Si	No

La elección de dichas frecuencias mínimas y esquemas de preprocesamiento se justifica puesto que para dichos valores se consiguieron mejores Fscore para los escenarios 1 a 5 como se muestra en el capítulo 3.2.

3.1.4. Adaptación del modelo final

Se presume que los motivos de llamada de los clientes al call center evolucionan con el paso del tiempo, consecuentemente se vuelve necesaria la actualización periódica del modelo.

Una vez elegido aquel modelo que mejor desempeña dentro de todos los casos propuestos en el capítulo 3.1.3 y habiéndose entrenado el mismo con una mayor cantidad de datos, se procederá a generar una arquitectura capaz de actualizar el mismo conforme lleguen nuevos requerimientos del call center.

En este contexto, se volverá vital experimentar sobre los parámetros que controlan el algoritmo OLDA, siendo estos el valor k y τ_0 que controlan cuan rápido se olvida la información antigua de entrenamiento y el offset que quita peso a las primeras muestras entregadas al modelo para su entrenamiento respectivamente.

Las guías a seguir para la búsqueda de los parámetros que optimicen este proceso se basan en las afirmaciones de los autores:

> *Higher values of the learning rate k and the downweighting parameter τ_0 lead to better performance for small mini-batch sizes S, but worse performance for larger values of S. Mini-batch sizes of at least 256 documents outperform smaller mini-batch sizes.*[8]

Se definen así los parámetros para actualizar el modelo con los valores chunksize = 256, $\tau = 64$ y $k = 0.5$.

3.2. Resultados

3.2.1. Base de clientes con suscripción

Se presentan en este capítulo los resultados obtenidos para la implementación del modelo LDA como clasificador sobre la base de clientes suscritos según los cinco escenarios de preprocesamiento presentados en la Tabla 3.3.

3.2.1.1. Escenario 1

Para el desarrollo del escenario 1 se realizó el preprocesado general de los requerimientos (i.e caracteres especiales y *stopwords*) y posteriormente se realizó *stemming*.

Una vez realizado el preprocesado, se tuvo que de los 320 modelos a explorar solo 240 tuvieron un número distinto de palabras en el vocabulario resultante. Es decir, para 80 de los modelos a estudiar ocurrió que su cantidad total de palabras en diccionario ya había ocurrido en un modelo anterior, por lo que los resultados serían los mismos.

Como resultado a la ejecución del algoritmo 3 se obtuvo el siguiente diccionario para esta configuración de escenario.

Tópico	p1	p2
Facturación	document	bolet
Datos	naveg	-
Voz	llam	reclam
Otros	pag	entel

Tabla 3.4: Diccionario generado para escenario 1, stemming y 4 tópicos.

Aun cuando la Tabla 3.4 es acotada en cuanto a la cantidad de palabras, la representatividad que estas tuvieron para cada uno e los tópicos permitió una buena asignación de nombres.

El desempeño específico de los modelos en este escenario se presenta en la Tabla 3.5.

Métrica	Máximo	Mínimo	Promedio	Desviación estándar
Fscore	0,724	0,428	0,652	0,056
Top2v2	0,771	0,484	0,700	0,046

Tabla 3.5: Desempeño general escenario número 1 de preprocesamiento.

De forma general, se aprecia que este escenario presenta alta desviación estándar para el desempeño de sus modelos conforme la frecuencia mínima de palabras exigidas variaba.

Se destaca que el desempeño máximo para la métrica Fscore y Top2v2 no se obtuvo bajo el mismo valor de frecuencia mínima exigida. Por otra parte, con valores de 0,428 y 0,484 para el Fscore y Top2v2 respectivamente, el escenario 1 presenta los peores mínimos desempeños de entre todos los escenarios de preprocesamiento propuestos.

Se presentan a continuación los cambios producidos en el tamaño del vocabulario de entrenamiento a medida que se exigía una frecuencia mayor de las palabras en el corpus, dicha evolución se presenta en la Figura 3.3.

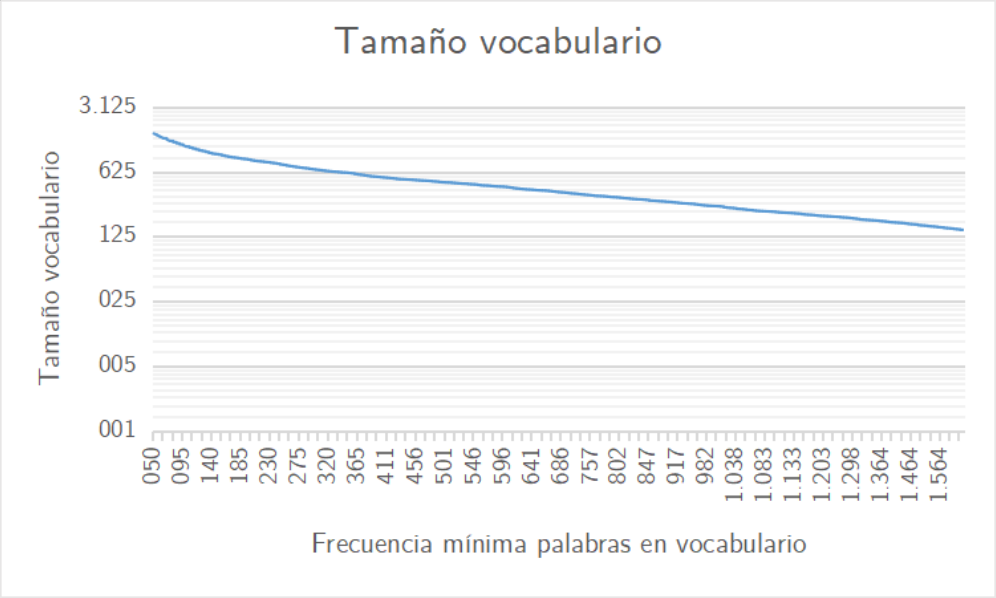

Figura 3.3: Tamaño vocabulario escenario 1, base de clientes suscritos.

Globalmente, los resultados obtenidos para este escenario se muestran en la Figura 3.4, se remarca la baja estabilidad del modelo conforme se varía la frecuencia mínima de las palabras en el diccionario.

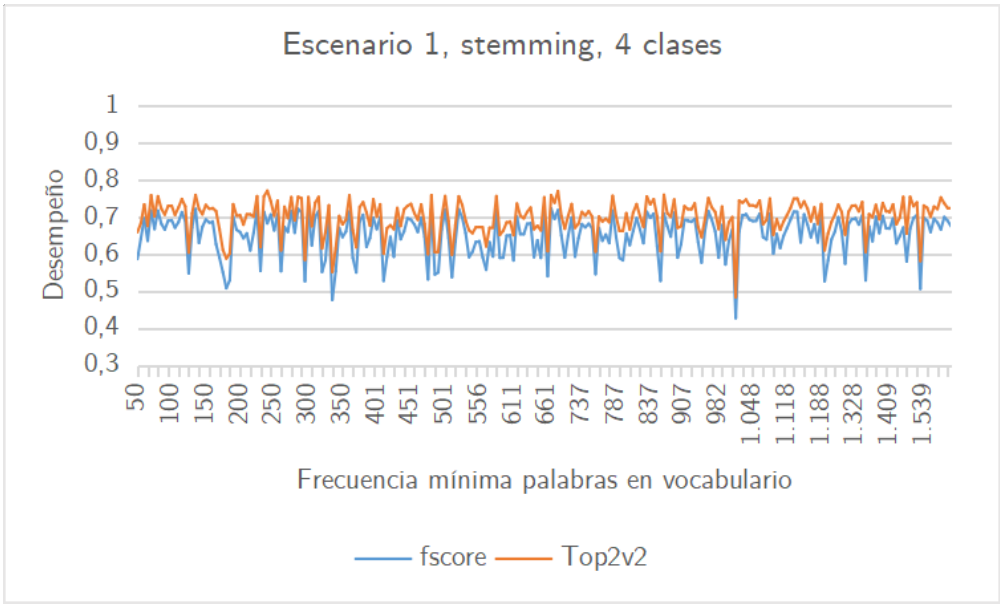

Figura 3.4: Métricas de desempeño Fscore y Top2v2 escenario 1.

3.2.1.2. Escenario 2

Para el desarrollo del escenario 2 se realizó el preprocesado general de los requerimientos y posteriormente se generaron bigramas y trigramas sobre el corpus y finalmente se aplicó lematización

sobre el mismo.

Una vez realizado el preprocesamiento del corpus, se tuvo que del total de 320 modelos a explorar, 251 generaron diccionarios finales de palabras con distinto largo. El diccionario resultante para la asignación de nombres en este escenario se presenta en las tablas 3.6 y 3.7.

Tópico	p1	p2	p3	p4
Facturación	documentar	boleta	impugnación_documentación_boleta_nro	cobrar
Datos	no_navegar	internet	navegación	plan
Voz	no_poder	llamar	servicio	hacer
Otros	entel	indicar	realizar	hacer

Tabla 3.6: Diccionario generado para escenario 2, n-gramas y lematización para 4 tópicos (parte 1).

Tópico	p5	p6	p7	p8	p9
Facturación	-	-	-	-	-
Datos	red	cobertura	gb	-	-
Voz	cobertura	sector	señal	-	-
Otros	plan	cambiar_plan	servicio	no_poder	spotify

Tabla 3.7: Diccionario generado para escenario 2, n-gramas y lematización para 4 tópicos (parte 2).

La versión final del diccionario para asignación de nombres es considerablemente más grande que en el escenario anterior, el tópico de "facturación"se caracterizó con menos palabras principalmente por la fuerte y exclusiva relación de ellas con el tópico en sí.

Por otra parte, el formato al cual se convergió con el diccionario para asignar de mejor forma los tópicos conllevó a que tanto requerimientos como servicios de valor agregado convergieran al tópico Otros, comportamiento que se presumió ocurriría.

El desempeño específico de los modelos estudiados se resume en la Tabla 3.8.

Métrica	Máximo	Mínimo	Promedio	Desviación estándar
Fscore	0,713	0,518	0,635	0,047
Top2v2	0,781	0,599	0,719	0,045

Tabla 3.8: Desempeño general escenario número 2 de preprocesamiento.

Inicialmente no se afirma una completa mejoría con respecto al escenario 1 pues este modelo posee un máximo desempeño en Fscore menor con respecto a su predecesor pero si obtiene un mejor desempeño para la métrica Top2v2.

Contrastando lo anterior, este escenario presenta un desempeño mínimo tanto en Fscore como en Top2v2 sustancialmente mayor al escenario anterior, siendo así todos los desempeños de modelos pertenecientes a este esquema mayores a 0,518 y 0,599 para las métricas de interés respectivamente.

Por último, se remarca una mejoría en cuanto a las desviación estándar calculada para los desempeños de los modelos, presentando así valores de 0,047 y 0,045 respectivamente para las métricas Fscore y Top2v2.

La totalidad de los modelos estudiados en este escenario se presentan en la Figura 3.5, estos resultados son consistentes con el escenario anterior, baja estabilidad conforme se varía la frecuencia mínima de palabras en el diccionario y estrecha relación entre ambas métricas.

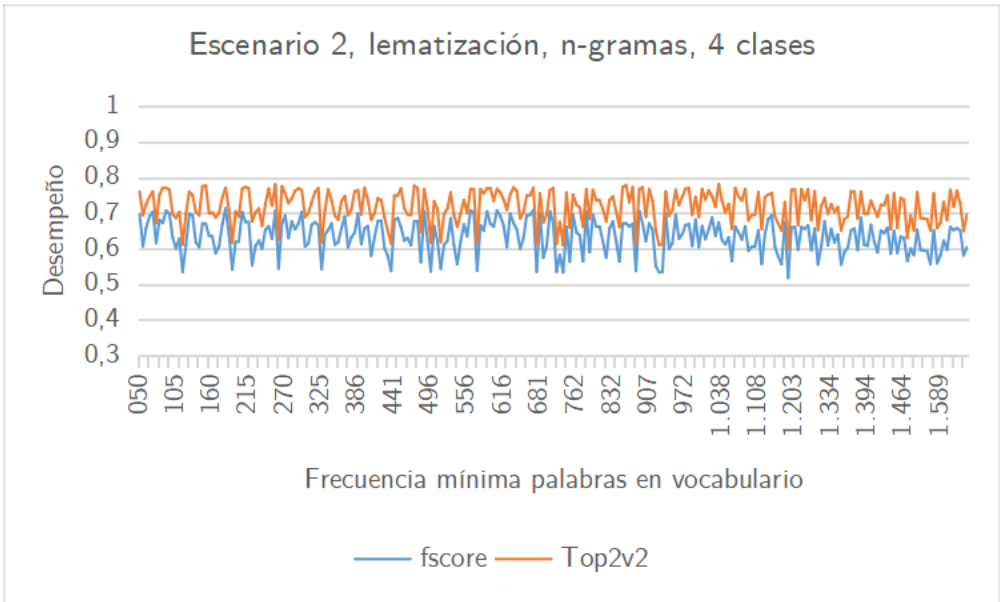

Figura 3.5: Métricas de desempeño Fscore y Top2v2 escenario 2.

3.2.1.3. Escenario 3

Este escenario tomó las mismas bases de preprocesamiento del escenario 2, sin embargo, al momento de entrenar el modelo LDA se le solicitó caracterizar el corpus para 5 tópicos generativos en lugar de 4.

Por su parte, el diccionario de tópicos se mantuvo intacto con respecto al señalado en las tablas 3.6 y 3.7.

La principal motivación para estudiar este escenario fue probar la capacidad de que el modelo discriminase un nuevo tópico de reclamos correspondiente a un subconjunto de la clase 'Otros' (e.g canales, servicio técnico o requerimientos).

Al evaluarlo, se evidenció un mejor desempeño trabajando con un diccionario de solo 4 clases, esto indica que una de las clases previamente existente es caracterizable en dos formas distintas, efecto causado posiblemente por patrones en la forma de escribir de los ejecutivos del call center.

En consecuencia y como ejemplo genérico, un requerimiento podría ser caracterizado bajo la clase 0 mientras que otro requerimiento podría ser clasificado como la clase 5, clases distintas para el modelo pero finalmente etiquetadas de igual forma en términos de una etiqueta legible.

Los resultados específicos obtenidos para este escenario se presentan en la Tabla 3.9.

Métrica	Máximo	Mínimo	Promedio	Desviación estándar
Fscore	0,716	0,504	0,641	0,037
Top2v2	0,765	0,582	0,701	0,034

Tabla 3.9: Desempeño general escenario número 3 de preprocesamiento.

La principal mejora con respecto al escenario 2 consistió en la disminución de cada una de las desviaciones estándares asociadas en un 1 %, dicho aspecto es notable en la medida que entrega más estabilidad frente a los parámetros del modelo.

Hubo también una mejoría de 0,03 para el máximo Fscore con respecto al escenario 2, sin embargo, está junto a las otras diferencias no son suficientemente notables como para considerar a este modelo totalmente superior.

Los resultados globales de este escenario se presentan en la Figura 3.6, al igual que en escenarios anteriores, el desempeño del modelo fue poco estable en función de la frecuencia mínima de palabras en vocabulario, además, se aprecia una disminución en la diferencia promedio del Fscore y la métrica Top2v2.

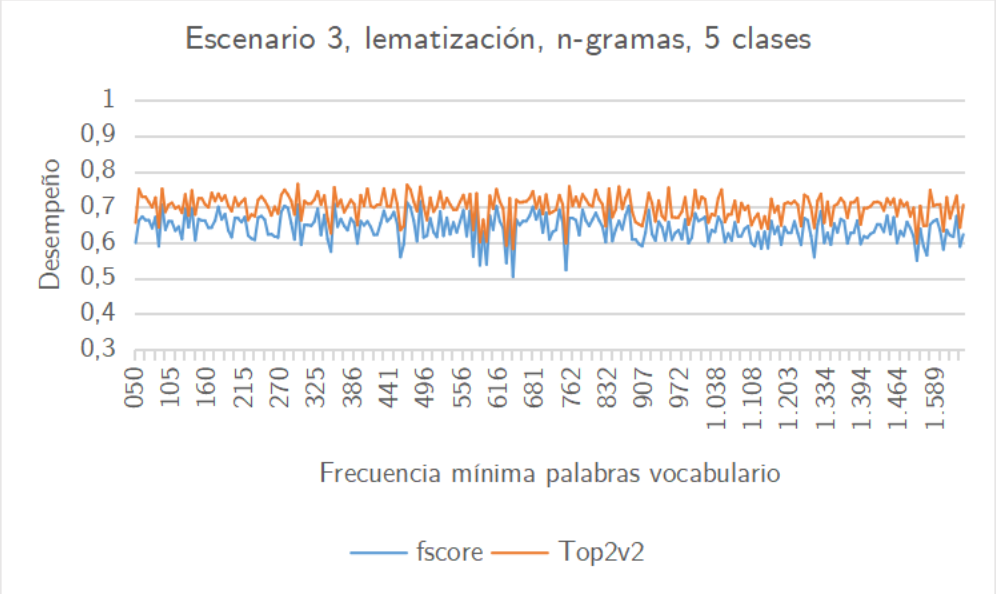

Figura 3.6: Métricas de desempeño Fscore y Top2v2 escenario 3.

3.2.1.4. Escenario 4

Este escenario consistió en la misma configuración de preprocesamiento que los escenarios 2 y 3 con la diferencia de que el modelo habría de caracterizar la generación del corpus a través de 6 tópicos distintos.

Al igual que en el escenario 3, el desempeño del escenario 4 fue superior en la medida que se implementó el diccionario de asignación de tópicos con solo 4 categorías.

Los resultados generales para Fscore y métrica Top2v2 del escenario 4 se presentan en la Tabla 3.7.

Métrica	Máximo	Mínimo	Promedio	Desviación estándar
Fscore	0,728	0,541	0,647	0,030
Top2v2	0,777	0,579	0,700	0,031

Tabla 3.10: Desempeño general escenario número 4 de preprocesamiento.

Se observa en la Tabla 3.10 como el escenario 4 presenta mejoría en todas las métricas con respecto a sus predecesores, salvo para el valor mínimo Top2v2.

De los escenarios estudiados, éste presenta las menores desviaciones estándares, dando lugar a una estabilidad un poco mayor con respecto a la frecuencia mínima de palabras en el diccionario que contiene el total de palabras en el corpus.

En la Figura 3.7 se muestran los resultados globales para las métricas de estudio en este escenario, al igual que en el escenario de estudio 3, la diferencia entre las métricas Fscore y Top2v2 es reducida.

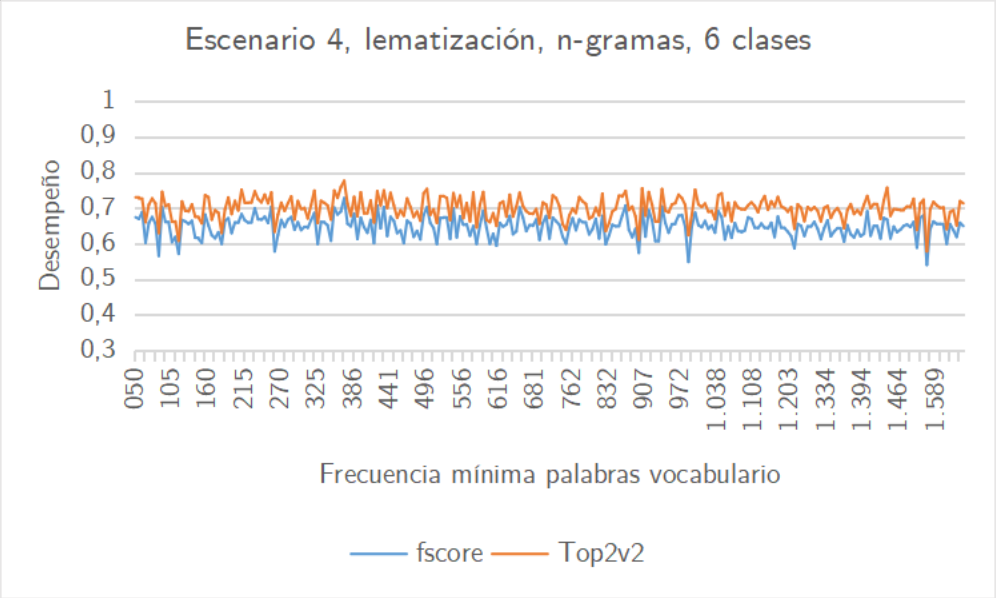

Figura 3.7: Métricas de desempeño Fscore y Top2v2 escenario 4.

3.2.1.5. Escenario 5

En este escenario de estudio se aplicaron todas las técnicas de preprocesamiento propuestas en el esquema de la Figura 3.2. Es decir, preprocesado básico, n-gramas, lematización y stemming.

Como resultado de ello, se debió crear un nuevo diccionario de asignación de nombres a tópicos, dicho diccionario es presentado en las tablas 3.11 y 3.12.

Tópico	p1	p2	p3	p4
Facturación	document	bolet	impugnación_documentación_boleta_nr	cobr
Dato	no_naveg	internet	naveg	plan
Voz	no_pod	llam	servici	hac
Otros	entel	indic	dificult	realiz

Tabla 3.11: Diccionario generado para escenario 5, n-gramas, lematización y stemming para 4 tópicos (parte 1).

Tópico	p5	p6	p7	p8	p9	p10	p11
Facturación	-	-	-	-	-	-	-
Dato	red	cobertur	gb	señal	sector	-	-
Voz	cobertur	sector	señal	-	-	-	-
Otros	hac	plan	cambio_pl	servici	ingres	no_pod	spotify

Tabla 3.12: Diccionario generado para escenario 5, n-gramas, lematización y stemming para 4 tópicos(parte 2).

Pese a que la única diferencia entre éste y el escenario anterior, fue la adición de stemming, el diccionario mostrado en las tablas 3.11 y 3.12 se modificó añadiéndose nuevas palabras, esto se debió a que añadiendo más técnicas de preprocesado el vocabulario global disminuyó, por ende, aumento la dificultad de diferenciar los temas 'Datos', 'Voz' y 'Otros'.

Del total de 320 modelos a estudiar, se tuvo un total de 251 escenarios únicos en términos de largo del diccionario de palabras, el resumen de desempeño de ellos se presenta en la Tabla 3.13.

Métrica	Máximo	Mínimo	Promedio	Desviación estándar
Fscore	0,723	0,471	0,651	0,045
Top2v2	0,795	0,575	0,739	0,043

Tabla 3.13: Desempeño general escenario número 5 de preprocesamiento.

De la tabla anterior, se destaca como este escenario presentó el mayor puntaje en métrica Top2v2 con un valor de 0,795. Por otra parte, los valores de desviación estándar entre modelos aumentaron con respecto a los escenarios con un mayor número de tópicos pero siguen manteniéndose dentro de un rango esperado.

Finalmente, se muestran los resultados globales para este escenario en la Figura 3.8. Se observa como la diferencia promedio entre las métricas estudiadas aumenta con respecto a escenarios que exploran un mayor número de tópicos.

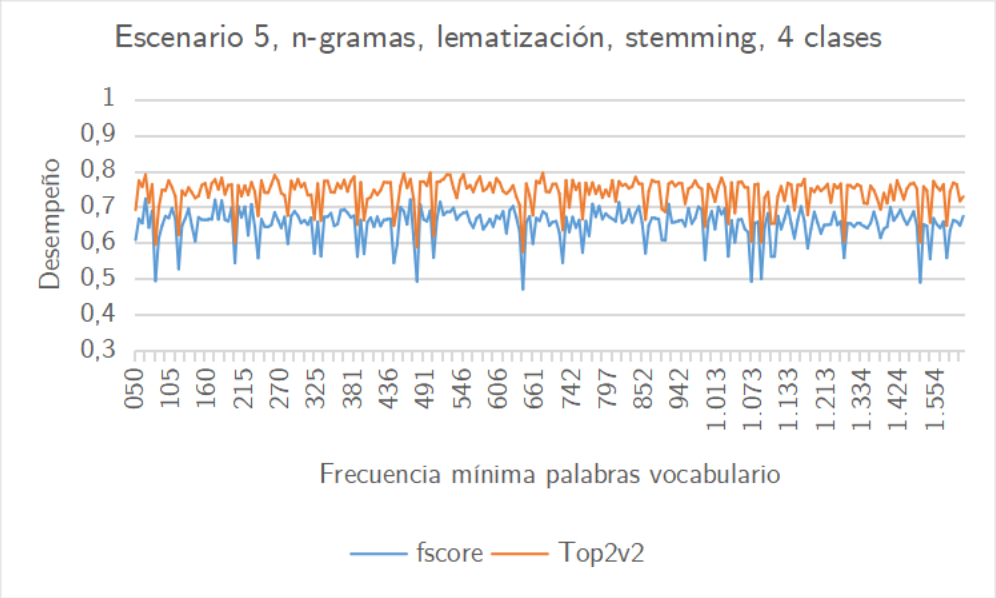

Figura 3.8: Métricas de desempeño Fscore y Top2v2 escenario 5.

3.2.1.6. Resumen

En términos generales, el desempeño obtenido por los modelos se resume en la Figura 3.9.

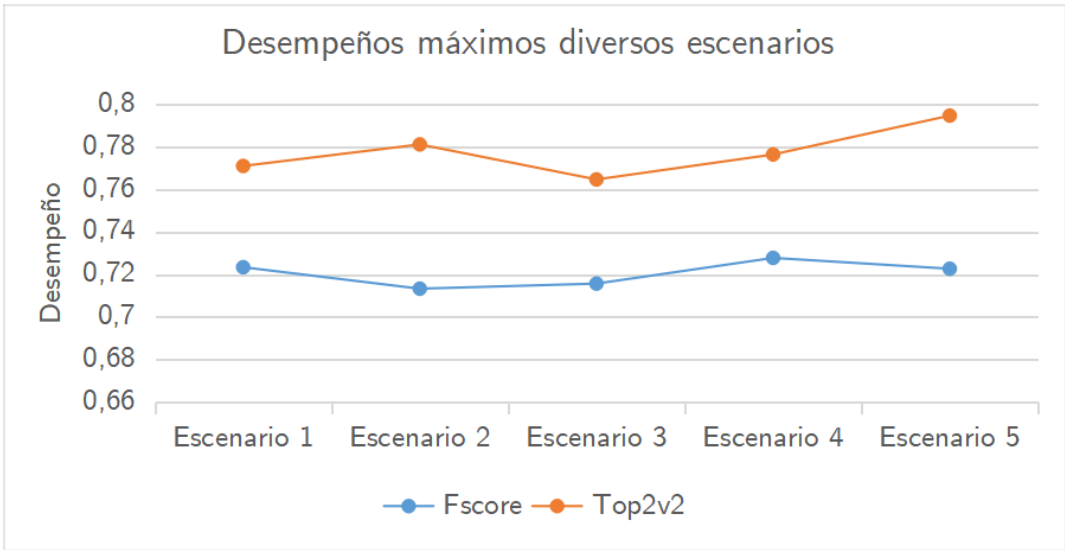

Figura 3.9: Desempeño máximo diversos escenarios.

Se tuvo así una métrica de Fscore variante para cada escenario, siendo dichas variaciones menores a las presentadas en la métrica Top2v2. Además, los escenarios 3 y 4 tuvieron mejores Fscore que su contraparte con mismo procesamiento (escenario 2), sin embargo dicha mejoría no se transmitió a la métrica Top2v2.

Las tablas 3.14 y 3.15 muestran los valores obtenidos en los mejores modelos probados dentro de cada uno de los escenarios de estudio, en ella se muestran también la cantidad de palabras que compuso el vocabulario para cada uno de los mejores modelos.

Métrica	Escenario 1	Escenario 2	Escenario 3	Escenario 4	Escenario 5
Fscore	0,723	0,713	0,715	0,727	0,722
Tamaño vocabulario	360	180	481	365	65

Tabla 3.14: Tamaño vocabulario para modelos de máximo Fscore en cada escenario.

Métrica	Escenario 1	Escenario 2	Escenario 3	Escenario 4	Escenario 5
Top2v2	0,771	0,781	0,764	0,764	0,794
Tamaño vocabulario	706	1007	295	365	496

Tabla 3.15: Tamaño vocabulario para modelos de máximo Top2v2 en cada escenario.

Tras implementar las diversas técnicas de preprocesamiento con los ajustes correspondientes de parámetros se opta por trabajar de ahora en adelante con el mejor modelo en Fscore del escenario 5.

Pese a que su supremacía con respecto a los mejores modelos del escenario 3 y 4 no fue plasmada en el Fscore, la cercanía a los mismos sin haber realizado la exploración solicitando un mayor número de tópicos al modelo da lugar a potenciales mejoras como trabajo futuro.

Tópico	precision	recall	Fscore	support
Datos	0,72	0,39	0,51	1161
Facturación	0,89	0,90	0,89	2616
Otros	0,53	0,70	0,60	781
Voz	0,49	0,68	0,57	708

Tabla 3.16: Reporte de clasificación mejor modelo LDA base de clientes suscritos.

Se evidencia en la Tabla 3.16 un muy buen desempeño del modelo en la categoría Facturación, sin embargo, dicho actuar no se condice en las demás categorías. Se observa un muy bajo valor de recall en la categoría de Datos, indicando una baja capacidad de identificar los requerimientos de dicha categoría como tal.

Para profundizar el quehacer del modelo se presenta a continuación la matriz de confusión del mismo, esta matriz se encuentra normalizada, en caso de desear identificar el total de muestras en cada celda se deberá ponderar el valor por el soporte asociado en la Tabla 3.16.

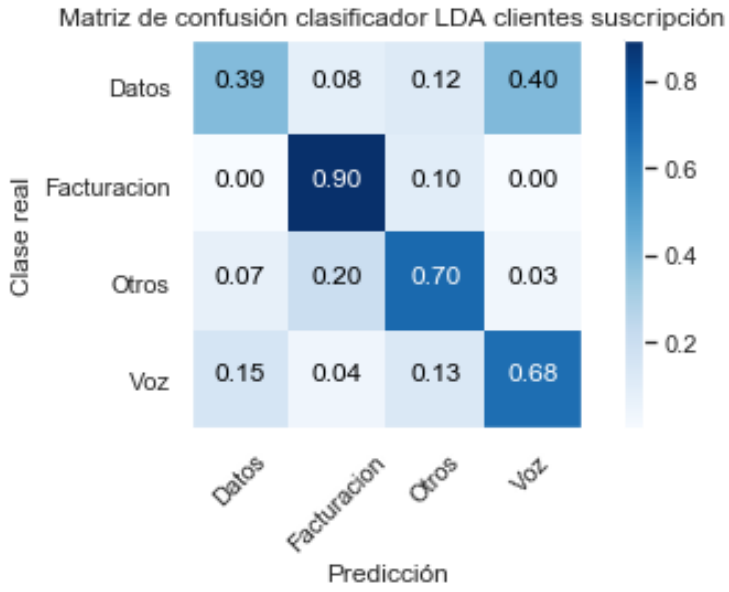

Figura 3.10: Matriz de confusión mejor clasificador LDA, base de clientes suscritos.

La matriz de confusión de la Figura 3.10 permite complementar los datos entregados por el reporte de clasificación anterior, se evidencia en ella como 2 de cada 3 requerimientos por problemas con servicio de datos son asociados a servicio de voz, comportamiento ad hoc tanto al recall de Datos como al precision de Voz.

En lo que respecta al resto de las categorías el modelo no presenta tendencias fuertes al momento de mal clasificar un requerimiento específico.

3.2.2. Base de clientes de prepago

En este capítulo se presentan los resultados obtenidos por el sistema clasificador LDA para la base de clientes de prepago.

Como consecuencia del buen desempeño el escenario para la base de clientes suscritos, se opta por trabajar exclusivamente con dicha técnica de preprocesamiento para distinta cantidad de tópicos por parte del modelo. Además, se mantiene el análisis de desempeño de los modelos en función de la frecuencia mínima exigida a las palabras para que pertenezcan al vocabulario final que toma en cuenta el modelo.

De esta forma, se exploran dos escenarios específicos para los clientes de prepago: un primer escenario en que se busca caracterizar la generación del corpus en base a 4 tópicos, y un segundo escenario de igual características que busque caracterizar el corpus con 6 tópicos (escenario 2).

3.2.2.1. Escenario 1

Habiendo incluido las técnicas de preprocesado básico, generación de n-gramas, lematizado y stemming. Se procedió a estudiar el desempeño de múltiples modelos en función de la frecuencia mínima a partir de la cual las palabras serían admitidas en el vocabulario que el modelo LDA tomaría en cuenta para entrenarse.

Debido a las grandes diferencias entre esta base de requerimientos y la presente en la base de clientes suscritos se iteró desde cero para llegar a un nuevo diccionario de asignación de nombres a los tópicos que retorna el modelo (ver tablas 3.17 y 3.18).

Tópico	p1	p2	p3	p4	p5
Voz	client	llam	lin	sald	ingres
Datos	naveg	comun	si	problem	cobertur
Otros	client	fech	app	compr	entreg
Bolsas y recargas	bols	recarg	sald	pes	client

Tabla 3.17: Diccionario generado para escenario 1, n-gramas, lematización y stemming para 4 tópicos (parte 1).

Tópico	p6	p7	p8	p9	p10
Voz	realiz	indic	plan	consult	verific
Datos	bols	senal	region	client	llam
Otros	realiz	llam	punt	solicit	dificult
Bolsas y recargas	compr	verific	mont	consum	ilimit

Tabla 3.18: Diccionario generado para escenario 1, n-gramas, lematización y stemming para 4 tópicos (parte 2).

Pese a la ausencia de bigramas o trigramas en las tablas anteriores, dichas palabras existieron

y se concentraron principalmente en la categoría 'Bolsas y recargas'.

El desempeño general de los modelos entrenados se presenta en la Tabla 3.19.

Métrica	Máximo	Mínimo	Promedio	Desviación estándar
Fscore	0,65	0,23	0,39	0,07
Top2v2	0,70	0,33	0,50	0,07

Tabla 3.19: Desempeño general escenario número 1 de preprocesamiento para base prepagos.

En primer lugar se puede establecer un desempeño menor en cuanto a Fscore y métrica Top2v2 para los máximos y mínimos del escenario. Se evidencia también una gran caída de desempeño general en comparación a los modelos de la base de clientes suscritos.

Dicha disminución en el desempeño general de los modelos estudiados se justifica debido a la naturaleza de la base la cual en términos generales tuvo menos y más breves requerimientos.

De esta forma, cuando se aumentó la frecuencia mínima para que el vocabulario tomara en cuenta las palabras, su cardinalidad disminuyó abruptamente, llegando a conjuntos de tan solo 6 palabras (ver Figura 3.11).

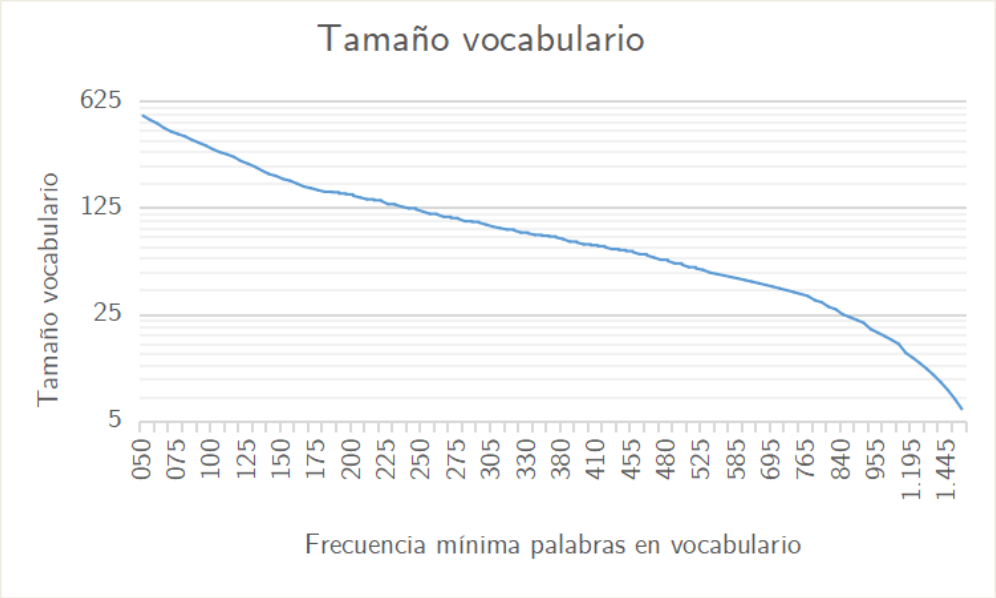

Figura 3.11: Tamaño vocabulario escenario 1, base de clientes de prepago.

En la Figura 3.11, el eje x toma en cuenta solo los valores que representaron un cambio real en la cantidad de palabras existentes en el vocabulario, i.e dada una frecuencia mínima (i), la frecuencia mínima (i+5) sería tomada en cuenta solamente si esta conllevaba una disminución en el total de palabras que componen vocabulario.

Se presentan a continuación los desempeños globales para el escenario 1 de la base de clientes de prepago.

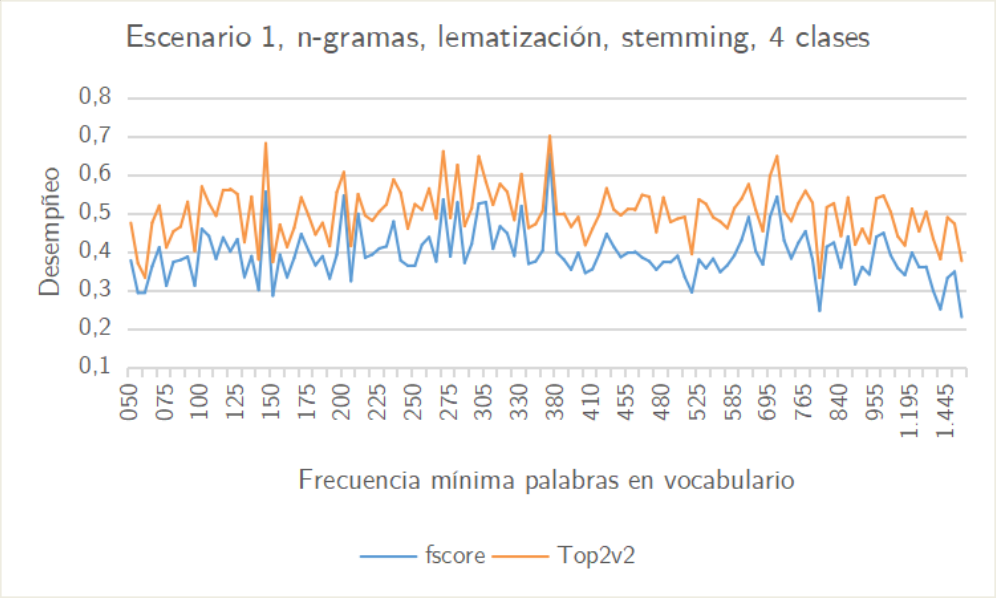

Figura 3.12: Métricas de desempeño Fscore y Top2v2 escenario 1 clientes de prepago.

Consecuentemente a lo comentado, se aprecia en la Figura 3.12 un bajo nivel de desempeño generalizado para los modelos salvo por un par de excepciones.

Además, se evidencia como a partir de una frecuencia mínima de 400, el desempeño en ambas métricas de interés tiende a la baja para todos los modelos, llegando hasta su mínimo en los casos de mayor frecuencia mínima.

3.2.2.2. Escenario 2

Al igual que en el escenario 1 de la base de clientes de prepago, en este escenario se aplicaron todas las técnicas de preprocesado sobre los requerimientos. El presente escenario se distingue del anterior puesto que ajusta el modelo para caracterizar la generación del corpus en función de 6 tópicos distintos.

Como resultado preliminar, tras el uso del algoritmo 3, se obtuvo que este escenario desempeña de mejor forma con un diccionario de asignación de nombres a tópicos con 6 posibles clasificaciones.

Dicho diccionario se presenta en las tablas 3.20 y 3.20. Comparando los resultados con los tópicos originales del escenario 1 se evidencian la división del motivo 'Bolsas y recargas' en sus dos componentes 'Bolsas' y 'Recargas'. Además, surgen los motivos de reclamo 'Canales' y 'Solicitudes' ambos anteriormente asociados al conglomerado del tópico 'Otros'

Tópico	p1	p2	p3	p4	p5
Bolsas	bols	compr	ilimit	consum	bolsa_mb
Recargas	recarg	sald	pes	lleg	client
Voz	problem	llam	comun	client	señal
Datos	naveg	si	equip	no_naveg	bols
Solicitudes	client	realiz	llam	sald	plan
Canales	ingres	cliente	app	lug	acced

Tabla 3.20: Diccionario generado para escenario 2 de base prepagos, n-gramas, lematización y stemming para 4 tópicos(parte 1).

Tópico	p6	p7	p8	p9	p10
Bolsas	servici	-	-	-	-
Recargas	verific	mont	medi	-	-
Voz	cobetur	indic	hac	region	-
Datos	client	cobertur	red	rrss	bam
Solicitudes	consult	solicit	lin	no_pod	indic
Canales	portal	aparec	registr	movil	mensaje_error

Tabla 3.21: Diccionario generado para escenario 2 de base prepagos, n-gramas, lematización y stemming para 4 tópicos (parte 2).

En términos específicos, el diccionario generado requirió de menor cantidad de palabras específicas para caracterizar de buena forma los tópicos asociados exclusivamente a dinero o adquisición de servicios. Por otra parte, aquellos tópicos asociados a un mayor número de causas posibles requirieron de un mayor número de palabras características.

Tras realizar las predicciones se hicieron dos transformaciones: las clasificaciones de 'Solicitudes' y 'Canales' se transformaron a 'Otros' y las clasificaciones de 'Bolsas' y 'Recargas' como 'Bolsas y recargas'.

Esta acción es netamente una medida para facilitar la comparación entre las distintas bases, sin embargo, el modelo queda disponible para realizar su quehacer en la forma para la cual fue originalmente diseñado.

Los resultados específicos obtenidos para Fscore y Top2v2 se presentan en Tabla 3.22.

Métrica	Máximo	Mínimo	Promedio	Desviación estándar
Fscore	0,71	0,32	0,50	0,10
Top2v2	0,74	0,40	0,58	0,08

Tabla 3.22: Desempeño general escenario número 2 de preprocesamiento para base prepagos.

Los altos valores de las desviaciones estándares observados en la Tabla 3.22 se justifican principalmente por la reducción del tamaño del vocabulario del modelo conforme se aumentaba la frecuencia mínima de las palabras en el corpus para pertenecer a él (Figura 3.11).

Aun cuando los desempeños generales son menores a los observados en los escenarios de la base de clientes suscritos, los valores máximos alcanzados para este escenario representan un modelo aceptable en desempeño. A continuación se presentan el desempeño global de los modelos del escenario 2.

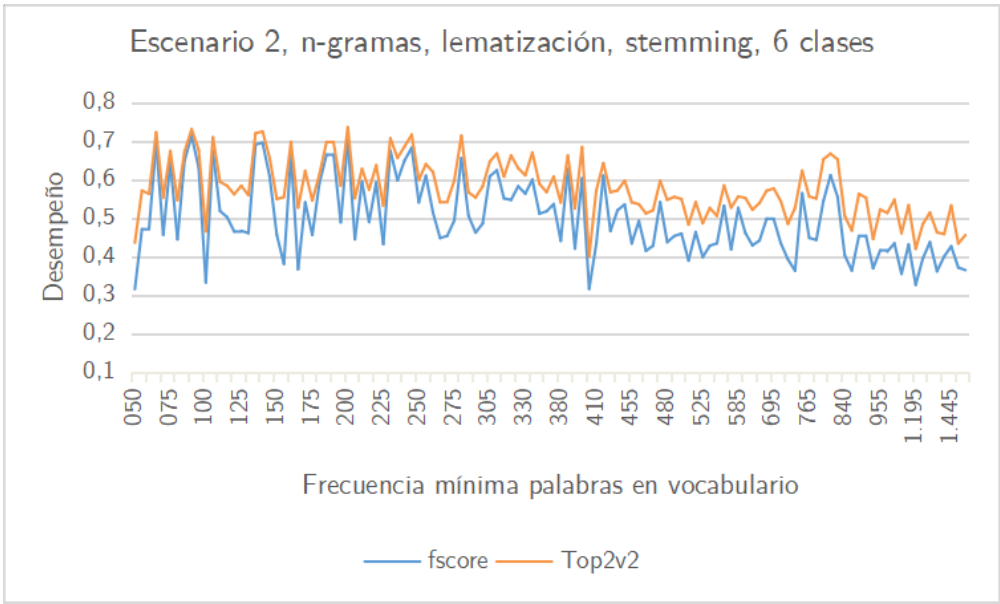

Figura 3.13: Métricas de desempeño Fscore y Top2v2 escenario 2 clientes de prepago.

En la Figura 3.13 se observa la disminución de desempeño conforme disminuye el tamaño de vocabulario. Y también, la gran similitud entre Fscore y Top2v2 de entre todos los casos de estudio.

3.2.2.3. Resumen

De forma general se resumen a continuación los mejores desempeños obtenidos para ambos escenarios de la base de prepagos en la Figura 3.14.

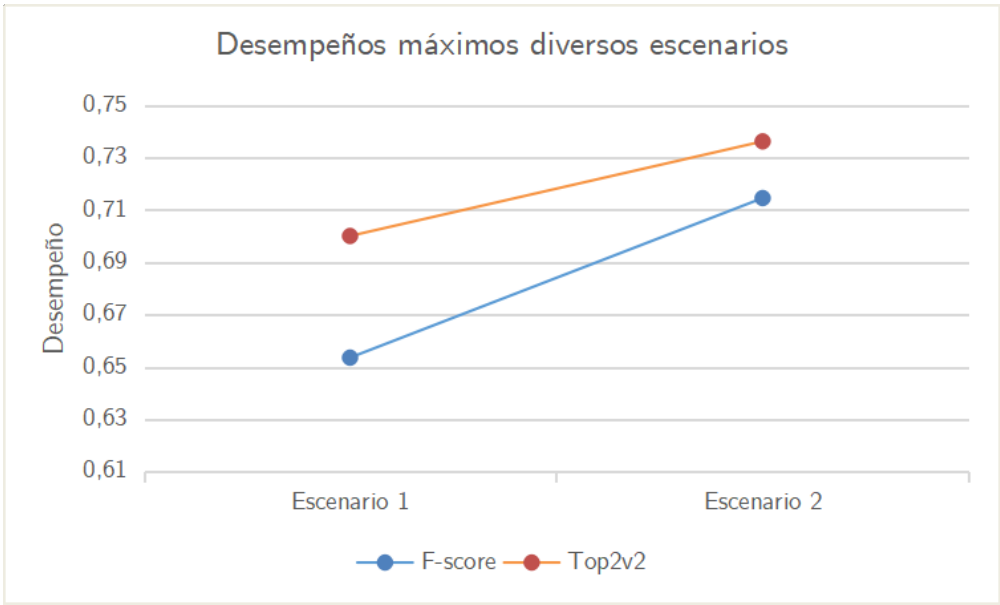

Figura 3.14: Desempeño máximo escenarios estudiados base clientes de prepago.

Se aprecia en la Figura 3.14 como el escenario 2 de estudios supera en ambas métricas de interés al primer escenario, consecuentemente se opta por mantener el mejor modelo del segundo escenario como el mejor modelo.

El detalle específico de desempeño de los mejores modelo se presenta en las siguientes tablas.

Métrica	Escenario 1	Escenario 2
Fscore	0,654	0,715
Tamaño vocabulario	80	333

Tabla 3.23: Tamaño vocabulario para modelos de máximo Fscore en cada escenario.

Métrica	Escenario 1	Escenario 2
Top2v2	0,700	0,736
Tamaño vocabulario	80	150

Tabla 3.24: Tamaño vocabulario para modelos de máximo Top2v2 en cada escenario.

En las tablas anteriores se plasma nuevamente la superioridad del mejor modelo del escenario 2 por sobre su contraparte del escenario 1, se observa además una tendencia de todos los mejores escenarios a tener un tamaño de vocabulario reducido, caso contrario a lo ocurrido para la base de clientes suscritos.

Se destaca como estos escenarios de preprocesamiento representaron las mayores similitudes entre ambas métricas de interés, dicho comportamiento indica que al añadir una segunda predicción, no se obtiene una mejoría sustancial.

A continuación se presenta el desempeño del mejor modelo clasificador LDA para esta base de clientes. El detalle del desempeño del mismo es presentado en la Tabla 3.25 y en la Figura 3.15.

Tópico	precision	recall	Fscore	soporte
Bolsas y recargas	0,80	0,84	0,82	2.001
Datos	0,79	0,53	0,63	1.088
Otros	0,77	0,63	0,69	1.417
Voz	0,49	0,80	0,60	813

Tabla 3.25: Reporte de clasificación mejor modelo LDA base de datos de prepago.

Se evidencia en la Tabla 3.25 buenos valores de precision salvo para la clase Voz, en ella aproximadamente un 50 % de los requerimientos clasificados como tal correspondieron realmente a problemas del servicio de voz.

Además, el modelo muestra buenos desempeños en recall salvo en la clase 'Datos', donde tan solo el 50 % de las muestras pertenecientes a ella fueron identificadas correctamente.

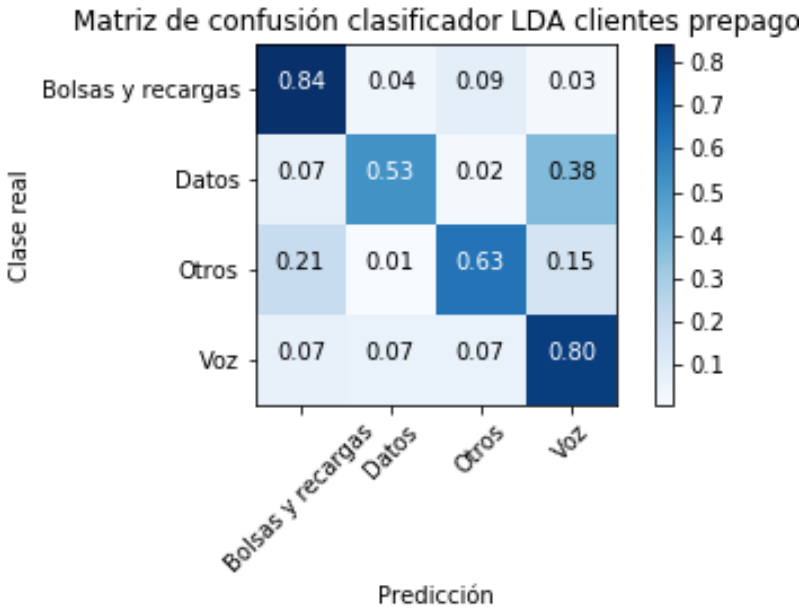

Figura 3.15: Matriz de confusión mejor clasificador LDA, base de clientes prepago.

La matriz de la Figura 3.15 permite identificar las falencias señaladas en el análisis del reporte de clasificación del modelo e indagar en las mismas. En primer lugar, se mal clasificó un 38 % del total de muestras de la clase Datos en la clase Voz, provocando una disminución tanto en el recall de Datos como en precision de Voz.

En segundo lugar, para la categoría Otros, el modelo categorizó los errores principalmente como Bolsas y recargas o como Voz, explicando así las bajas de la métrica precision en dichas categorías.

3.3. Cierre primera metodología

Al finalizar la etapa inicial de construcción del sistema de análisis de tópicos para interacciones clientes-call center se concluye que el modelo LDA como clasificador cumple parcialmente los objetivos planteados; tanto para la base de clientes de prepago como para la base de clientes suscritos.

Ambos clasificadores generados presentaron sus máximos desempeños en Fscore ponderado (i.e promedio simple sin tomar en cuenta desbalance de clases) frente a esquemas completos de preprocesamiento y con vocabularios de tamaños no mayores a 400 palabras.

Los clasificadores creados presentaron dificultades en diferenciar principalmente los reclamos asociados a problemas de servicio de datos con aquellos asociados al servicio de voz. Dicho comportamiento es asociable a la gran zona etérea que comprenden los reclamos por cobertura.

Finalmente, se concluye por finalizada esta etapa del sistema de automatización con clasificadores LDA de desempeño en Fscore ponderado de 0,715 y 0,720 para las bases de clientes de prepago y suscripción respectivamente.

Capítulo 4

Mejoras sobre modelo resultante LDA

Tras finalizar la implementación del modelo clasificador LDA para ambas bases abordadas, se concluyó que hubo cumplimiento parcial de objetivos planteados. Dado el valor 0.72 de Fscore macro alcanzado por ambos modelos se presume un desempeño medio, sin embargo, potencialmente mejorable.

En el presente capítulo se detallan las acciones que permitieron mejorar el desempeño del sistema sobre la primera clase predicha. Dichas acciones se guiaron por 3 ideas principales:

1. Abordar el problema mediante un esquema de aprendizaje supervisado.
2. Utilizar los resultados de los modelos LDA en su forma de vector de probabilidad.
3. Utilizar información extra añadida por el ejecutivo al recibir las llamadas.

Utilizar los resultados del modelo LDA en su formato original permitirá tener nuevas variables predictoras al modelo nuevo. Al presentarse de esta forma, dichas variables serán capaces de entregar más información que con el uso dado en el capítulo 3.

Finalmente, el modelo resultante se creará aplicando todas las etapas involucradas en la generación y entrenamiento del mejor modelo LDA obtenido en el capítulo 3, el preprocesado de la información extra añadida por los ejecutivos para dar solución a los problemas de cada llamada, y además la etiqueta asignada manualmente.

4.1. Metodología

El esquema general de trabajo para esta capítulo se detalla en la Figura 4.1.

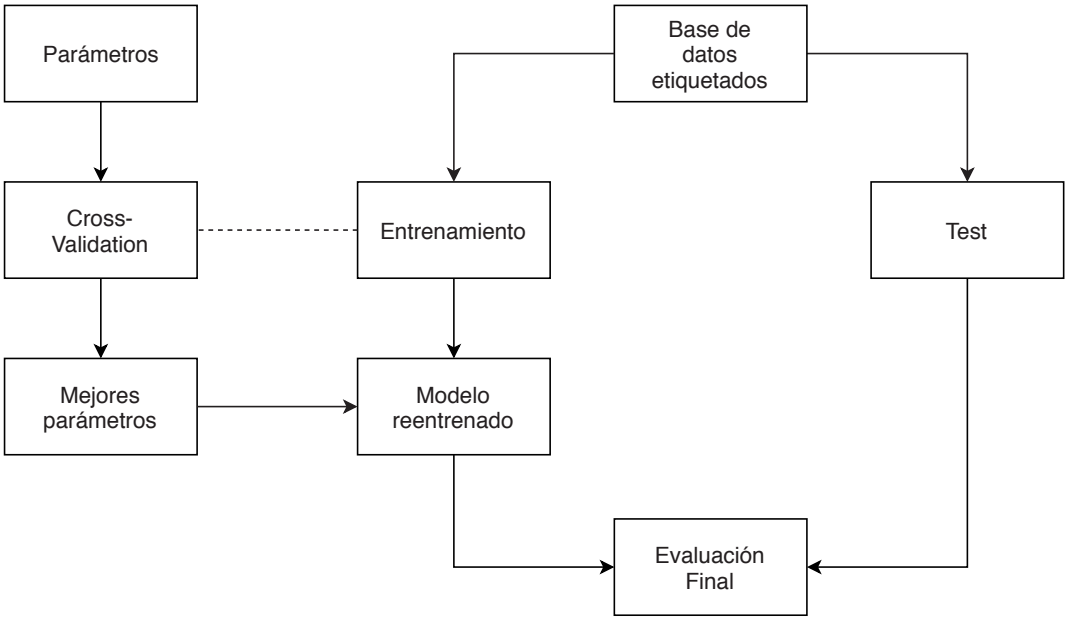

Figura 4.1: Esquema desarrollo algoritmos aprendizaje supervisado.

En primer lugar, se segmentaron las bases de datos (detalle en capítulo 4.1.1) en dos partes: un conjunto de entrenamiento (90 %) y un conjunto de evaluación (10 %). Los algoritmos de aprendizaje supervisado implementados en este esquema fueron: *Multi Layer Perceptron* (MLP), *Support Vector Machine* (SVM) de kernel gaussiano y además un *Random Forest* (RF).

Para encontrar los mejores parámetros de cada modelo se realizó una búsqueda de malla. En pro de evitar sobreajuste, dicha búsqueda se implementó bajo un esquema de *cross-validation* de 4 pliegues. Es decir, se realizaron 4 particiones distintas del conjunto de entrenamiento en dos subgrupos de 75 % y 25 % de las muestras respectivamente.

Una vez encontrados los parámetros con mejor desempeño promedio para el esquema cross-validation, se procedió a entrenar los modelos sobre el conjunto de entrenamiento (esta vez sin sub-particiones).

Finalmente, se comparó el desempeño de los diversos modelos, estudiando principalmente el Fscore obtenido puesto que no se tuvo prioridades específicas para la disminución de falsos negativos por sobre falsos positivos o viceversa.

4.1.1. Bases de datos

Las bases de datos a utilizar en la creación de este nuevo modelo corresponden a los conjuntos utilizados como validación para los modelos en el capítulo 3.

Con el objetivo de fortalecer el entrenamiento y evaluación del sistema se procedió a etiquetar 5.000 requerimientos nuevos para la base de clientes suscritos. Dicha acción se realizó únicamente en esa base de clientes.

Las nuevas variables predictoras correspondieron principalmente a categorías de acciones para solucionar los problemas de cada llamada recibida. Ellas fueron preprocesadas mediante un mapeo el cual las llevo desde su valor original a una etiqueta asociada a una de las posibles 4 clases de motivo de reclamo.

Dicho mapeo se realizó mediante un diccionario creado manualmente a partir de discusiones y opiniones expertas asociadas a los términos técnicos utilizados por los ejecutivos.

De esta forma, la base de clientes suscritos se compuso de aproximadamente 10.000 muestras mientras que la base de clientes de prepago de 5.000 muestras.

4.1.2. Escenarios a estudiar

4.1.2.1. Multilayer perceptron

Para los modelos MLP, se entrenaron modelos de dos capas ocultas. Cada uno de los escenarios a estudiar se diferenció de sus pares en el total de neuronas que poseyó por capa oculta.

Además, todos los modelos entrenados tuvieron una cantidad igual de neuronas en ambas capas ocultas. En total se estudiaron los desempeños para 50 modelos cuyos parámetros se señalan en la Tabla 4.1.

	Capa 1	Capa 2
N° de neuronas	1 -2 - 3 - ... - 30	1 -2 - 3 - ... - 30

Tabla 4.1: Parámetros a explorar MLP.

Además, la implementación realizada se caracteriza por: utilizar función de activación ReLu, usar método de optimización lbfgs y tener como salida de arquitectura 4 nodos respectivos a cada una de las posibles clases.

4.1.2.2. Random Forest

Para los modelos Random Forest se exploró la variación de desempeño en función de dos parámetros principales, la profundidad cada árbol y la totalidad de árboles dentro del bosque.

Específicamente, los modelos explorados fueron todas las todas las combinaciones posibles de realizar entre los valores detallados en la Tabla 4.2.

	N° de árboles	Profundidad
Valores	1 - 2 - 3 - ... - 50	1 - 2 - 3 - ... - 15

Tabla 4.2: Parámetros a explorar RF.

La razón tras el límite superior impuesto tanto para profundidad como cantidad de árboles del modelo se debió a que para mayores valores en dichos parámetros las mejoras serían mínimas en conjunto de entrenamiento.

Además, la implementación realizada se caracteriza por: exigir un mínimo de 2 muestras para realizar una partición válida, exigir un mínimo de una muestra por hoja para considerarla un nodo.

4.1.2.3. Support Vector Machine

La última serie de modelos a estudiar fue la de modelos SVM con kernel gaussiano (i.e RBF), para ello se exploraron las variaciones en desempeño al modificar los valores del parámetro de penalización por error (C) y el coeficiente gamma (γ) del kernel.

Específicamente, los modelos estudiados corresponden a todas las combinaciones posibles entre los valores de las columnas de la Tabla 4.3.

	C	γ
Valores	0,01 - 0,05 - 0,1 - 0,5 - 1 - 5 - 10	0,01 - 0,05 - 0,1 - 0,5 - 1 - 5 - 10

Tabla 4.3: Parámetros a explorar SVM.

Al igual que en los modelos anteriores, los estudios de este algoritmo no aplicaron técnicas para lidiar con el desbalance de clase.

4.2. Resultados

4.2.1. Base de clientes con suscripción

En este capítulo se presentan los resultados obtenidos para las implementaciones propuestas en el capítulo 3.1.3

4.2.1.1. Multilayer perceptron

Habiéndose entrenado los diversos modelos en modalidad de cross-validation para los diversos parámetros detallados, se obtuvo que los parámetros que maximizaron el Fscore ponderado del modelo fueron 10 neuronas en cada una de las capas ocultas.

Para dicha configuración, se obtuvo un valor de Fscore de 0,782 con desviación estándar de 0,007. La evolución de desempeño en función e la cantidad de neuronas por capa oculta se muestra en la Figura 4.2.

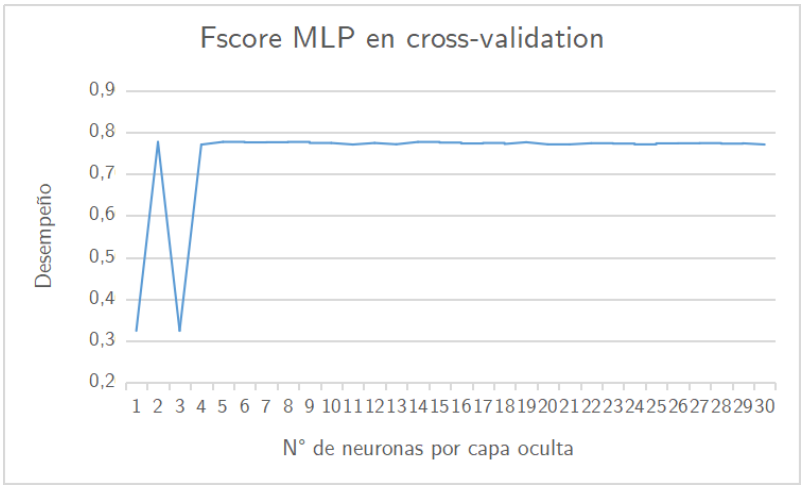

Figura 4.2: Fscore promedio en cross-validation para MLP clientes de suscripción.

Se observa en la Figura 4.2 una cota de desempeño en torno al valor 0,78, resultado ad hoc al máximo valor de Fscore obtenido, además, se opta por no aumentar aún más el número de neuronas por cada capa oculta para evitar sobreajuste.

El desempeño del mejor modelo en cross-validation aplicado al conjunto de validación es presentado en la Tabla 4.6. Se observa en ella un desempeño medio del modelo, aun cuando el modelo posee muy buenas capacidades para la categoría de facturación, es débil en las demás.

Tópico	precision	recall	fscore	support
Datos	0,74	0,73	0,73	260
Facturación	0,86	0,96	0,91	500
Otros	0,69	0,59	0,63	153
Voz	0,62	0,46	0,53	112

Tabla 4.4: Reporte de clasificación MLP en conjunto de evaluación, base clientes suscritos.

Finalmente, el modelo obtuvo un valor de Fscore ponderado de 0,781 en el conjunto de validación, métrica que oculta la discapacidad del mismo para clasificar con igual capacidad cada una de las clases, siendo en este caso, la categoría de voz la más débil.

4.2.1.2. Random forest

Al realizar la exploración de parámetros óptimos para el algoritmo RF se obtuvo un desempeño máximo para el Fscore ponderado con una profundidad de árbol de 8 niveles y un total de 30 estimadores.

Con dicha configuración, se obtuvo un desempeño en Fscore ponderado de 0,776 y una desviación estándar de 0,005. La evolución de desempeño en Fscore ponderado en función de los parámetros del modelo RF se detalla en la Figura 4.3.

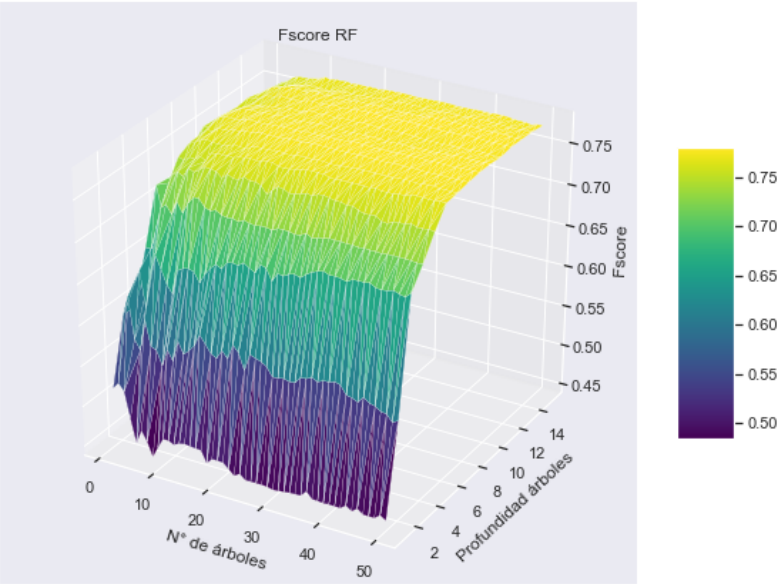

Figura 4.3: Fscore promedio en cross-validation para RF clientes de suscripción.

Se observa en la Figura 4.3 un bajo desempeño del modelo frente a árboles de profundidad menor a 5 niveles. Comportamiento que se repite para modelos con un total de 4 árboles o menos. Para el resto de los casos, el modelo tiene un estable desempeño en Fscore ponderado cercano a 0,75.

El desempeño en validación del mejor modelo RF es presentado en la siguiente Tabla.

Tópico	precision	recall	fscore	support
Datos	0,74	0,72	0,73	260
Facturación	0,86	0,96	0,91	500
Otros	0,69	0,61	0,65	153
Voz	0,62	0,43	0,51	112

Tabla 4.5: Reporte de clasificación RF en conjunto de evaluación, base clientes suscritos.

Se observa en la Tabla 4.5 un desempeño casi idéntico al modelo anterior (MLP) salvo por variaciones de hasta 2 puntos en algunas métricas. El desempeño promedio del modelo RF en Fscore es 0,7 sin embargo, ponderando por la cantidad de muestras en cada clase, dicha cifra sube a un valor de 0,781.

Se presume que este modelo también vio alterado su desempeño como consecuencia del desbalance de clases en la base, hecho que también se hace presente en la cantidad de muestras en cada categoría para la base de validación.

4.2.1.3. Support vector machine

Tras finalizar la exploración de parámetros óptimos para el modelo SVM-RBF se obtuvo que el mejor desempeño en cross-validation del modelo fue para valores de parámetro C = 10 y parámetro gamma = 0,5.

Con dichos parámetros, el modelo obtuvo en cross-validation un Fscore de 0,777 y una desviación estándar de 0,007. La evolución de desempeño del modelo según su Fscore ponderado en función de los parámetros estudiados se presenta en la Figura 4.7.

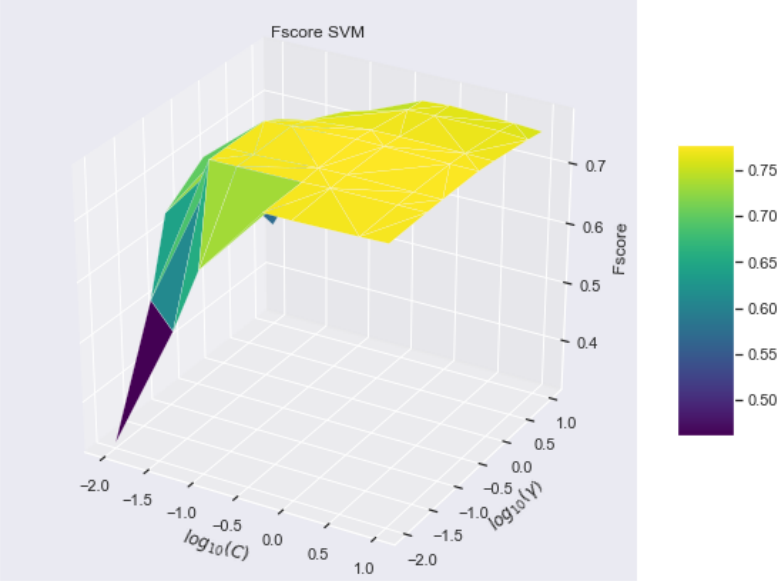

Figura 4.4: Fscore promedio en cross-validation para SVM clientes de suscripción.

Se observa en los resultados obtenidos un tope estable para valores de C mayores a 1 y parámetro gamma mayor a 0,05 pero menor a 10. Dicho tope tiene un valor aproximado en Fscore ponderado de 0,77. Los resultados específicos del mejor modelo SVM en el conjunto de validación son presentados en la Tabla a continuación.

Tópico	precision	recall	fscore	support
Datos	0,74	0,73	0,73	260
Facturación	0,85	0,96	0,91	500
Otros	0,68	0,58	0,62	153
Voz	0,64	0,43	0,51	112

Tabla 4.6: Reporte de clasificación SVM en conjunto de evaluación, base clientes suscritos.

Al igual que los modelos anteriores, el desempeño general es medio, teniendo como punto alto la categoría de facturación y como mayor debilidad la categoría voz.

Finalmente se calcula un Fscore ponderado de 0,777 para el conjunto de validación, cifra idéntica a la obtenida por el modelo en el esquema de cross-validation.

4.2.2. Base de clientes de prepago

Se presentan en este capítulo los resultados obtenidos para las implementaciones propuestas en el capítulo 4.1.2 para la base de clientes de prepago.

4.2.2.1. Multilayer perceptron

Tras realizar la exploración de parámetros vía cross-validation para el algoritmo MLP se obtuvo que los parámetros que maximizaron el Fscore promedio del modelo fueron 13 neuronas en cada una de las capas ocultas.

Para dicha configuración, se obtuvo un valor de Fscore de 0,867 con desviación estándar entre de 0,012. La evolución de desempeño en función de la cantidad de neuronas por capa oculta se muestra en la Figura 4.5.

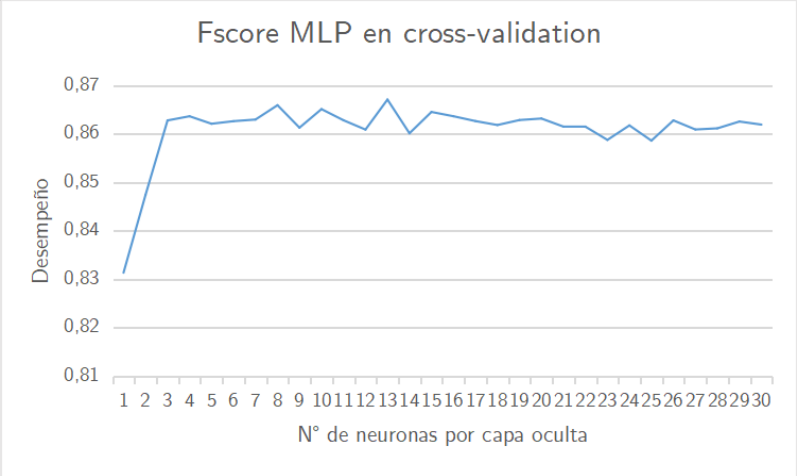

Figura 4.5: Fscore promedio en cross-validation para MLP clientes de prepago.

Se evidencia en la Figura 4.5 una cota de desempeño en torno al valor 0,86. Para prevenir sobreajuste se decidió no probar escenarios con mayor cantidad de neuronas.

El desempeño del en conjunto de validación para dicho modelo se presenta en la Tabla 4.7. Se observa en ella un buen desempeño del modelo el cual presento su menor capacidad de clasificación para los requerimientos de la categoría Voz con un recall del 75 %.

Tópico	precision	recall	fscore	soporte
Bolsas y recargas	0,88	0,92	0,90	195
Datos	0,83	0,83	0,83	100
Otros	0,87	0,92	0,90	149
Voz	0,92	0,75	0,83	88

Tabla 4.7: Reporte de clasificación MLP en conjunto de evaluación, base clientes prepagos.

Finalmente, el Fscore ponderado del modelo en el conjunto de validación fue de 0,875 indicando así su buen desempeño tanto en recall como en precision de cada una de las clases.

4.2.2.2. Random forest

Al realizar la exploración de parámetros óptimos para el algoritmo RF se obtuvo que los parámetros que maximizaron el Fscore promedio en cross-validation fueron una profundidad de 9 niveles y un total de 42 estimadores (i.e arboles).

Para dicha configuración, el modelo tuvo un desempeño en Fscore de 0,8710 asociado a una desviación estándar de 0,005.

La evolución de desempeño en términos del Fscore a medida que variaban los parámetros del modelo es presentada en la Figura 4.6.

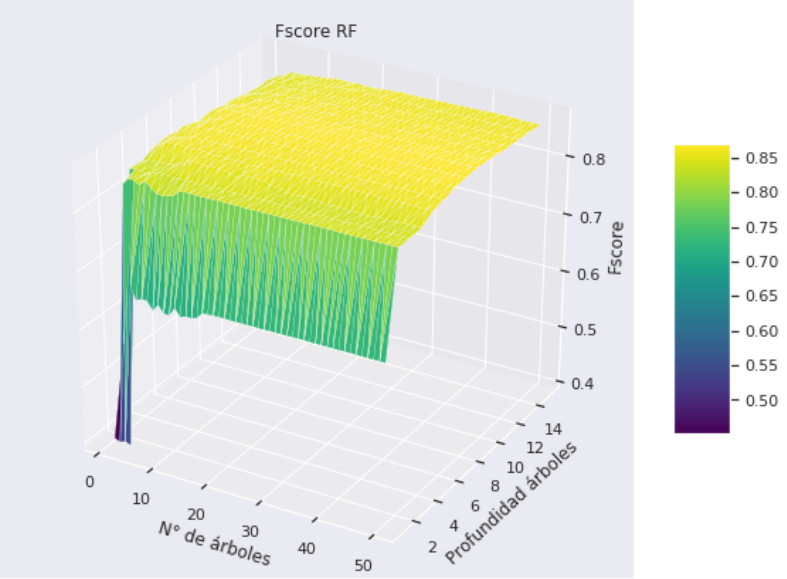

Figura 4.6: Fscore promedio en cross-validation para RF clientes de prepago.

En la Figura 4.6 se observa un comportamiento estable por parte del modelo salvo para los casos borde, siendo estos: un total de tres o menos estimadores, o una profundidad de menos de dos niveles.

El desempeño del mejor modelo RF para el conjunto de validación se muestra en la Tabla 4.8.

Tópico	precision	recall	fscore	soporte
Bolsas y recargas	0,90	0,92	0,91	195
Datos	0,85	0,82	0,84	100
Otros	0,88	0,94	0,91	149
Voz	0,89	0,77	0,83	88

Tabla 4.8: Reporte de clasificación RF en conjunto de evaluación, base clientes prepagos.

Los resultados presentes en la Tabla 4.8 muestran un desempeño bueno por parte del modelo el cual resulta ser marginalmente mejor al modelo MLP pero que sin embargo, presenta la misma dificultad al momento de clasificar requerimientos de la categoría Voz.

Finalmente, el Fscore ponderado obtenido por el modelo para el conjunto de validación fue de 0,883 resumiendo así su buen desempeño.

4.2.2.3. Support vector machine

Tras realizar la exploración de los mejores parámetros para el modelo SVM-RBF se obtuvo que el mejor desempeño vía cross-validation del modelo fue con un parámetro C = 10 y un valor de gamma = 0,5.

Con dichos parámetros, el modelo obtuvo en cross-validation un Fscore de 0,864 y una desviación estándar de 0,006. Valores cercanos a los obtenidos por el modelo RF y además indicadores de un buen desempeño del modelo.

La evolución de desempeño en Fscore en función de los parámetros estudiados se presenta en la Figura 4.7.

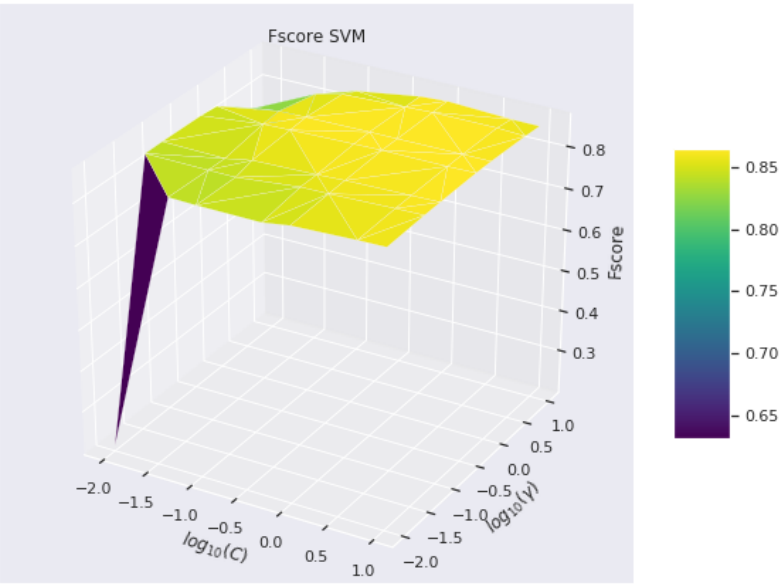

Figura 4.7: Fscore promedio en cross-validation para SVM clientes de prepago.

Se observa en los resultados obtenidos una fuerte estabilidad frente a cambios de parámetros exceptuando el caso borde en que tanto C como γ toman el valor 0,01.

El desempeño del mejor modelo SVM para el conjunto de validación se muestra en la Tabla 4.9.

Tópico	precision	recall	fscore	soporte
Bolsas y recargas	0,90	0,92	0,91	195
Datos	0,84	0,85	0,85	100
Otros	0,87	0,93	0,90	149
Voz	0,92	0,74	0,83	88

Tabla 4.9: Reporte de clasificación SVM en conjunto de evaluación, base clientes prepagos.

Se evidencia en la Tabla 4.9 el buen desempeño del modelo SVM para el conjunto de validación, al igual que sus pares, el modelo presenta su menor capacidad de desempeño en la categoría de requerimientos asociados al servicio de voz.

Finalmente, el Fscore ponderado obtenido por este modelo SVM en el conjunto de validación fue 0,882.

4.3. Cierre segunda metodología

Tras implementar la metodología planteada en este capítulo se concluye que se ha logrado mejoría de desempeño esperado en el sistema de automatización de interacciones.

El modelo de mejor desempeño se obtuvo con el algoritmo RF tanto para suscripción como para prepago utilizando configuraciones específicas de hiperparámetros en cada conjunto de datos. En ambos casos se maximiza el Fscore ponderado en el conjunto de validación.

Por una parte, el desempeño asociado al conjunto de clientes de prepago aumentó, llegando a un valor de Fscore ponderado de 0,883, mejorando en 17 puntos el puntaje de Fscore ponderado del clasificador LDA.

Por otra parte, para la base de clientes de suscripción el Fscore ponderado alcanzo el valor de 0,781, mejorando en 6 puntos el puntaje de Fscore ponderado obtenido por el modelo LDA clasificador para los clientes de suscripción.

Conclusiones

Conforme al paso de los años la humanidad se adentra cada vez más en la denominada era digital. Dicha era conlleva transformaciones en distintas industrias, los servicios que entregan y la forma en que lo hacen. Se identifica los call center como un objetivo específico de dichas transformaciones, con el objetivo de mejorar tanto la atención a clientes como el ahorro de costos, es menester trabajar la automatización de los mismos y la optimización de sus procesos.

Este trabajo consistió en la creación de un sistema capaz de identificar la razón de llamado de los clientes al call center de Entel en función de breves transcripciones de dichos llamados. El sistema desarrollado logra pronosticar dichas razones de buena forma para los requerimientos asociados a clientes de suscripción y de excelente forma para aquellos clientes de prepago.

En el caso de clientes de suscripción, el sistema desarrollado identifica la razón de llamada de entre cuatro posibles categorías: servicios de facturación, servicio de voz, servicio de datos u otros servicios. El desempeño final obtenido en este caso por el sistema se cuantifica con un Fscore ponderado de 0,78. Una apertura por cada uno de los motivos de llamada evidencia un claro desbalance de desempeño entre clases, retornando las mejores clasificaciones para requerimientos asociados a problemas de facturación (la clase más frecuente) y el peor desempeño para problemas por el servicio de voz (la clase menos frecuente).

En el caso de clientes de prepago, el sistema desarrollado también identifica la razón de llamada entre cuatro categorías: servicios de bolsas y recargas, servicio de voz, servicio de datos u otros servicios. Para este tipo de clientes, el desempeño final obtenido por el sistema se cuantifica con un Fscore ponderado de 0,88. En esta base de clientes el desempeño del sistema es consistente a través de la totalidad de las clases, echo asociado tanto al menor desbalance entre clases presentes en la base de datos y también a una mayor riqueza en la calidad de datos en la misma.

Como trabajo futuro, se identifica en primer lugar el deber de mantener en vigencia el sistema mediante la actualización de los modelos LDA respectivos a cada base, actividad realizable mediante el algoritmo OLDA sobre los nuevos requerimientos. Esta actividad toma especial importancia puesto que con el tanto la forma de redactar los problemas que aquejan a los clientes como sus naturalezas respectivas evolucionarán. Manteniendo el sistema actualizado se podrán generar mejores vectores de características para los algoritmos de aprendizaje supervisado respectivos.

Finalmente, para mejorar el desempeño del sistema en la base de clientes suscritos se propone dar uso a técnicas asociadas a sobre y sub muestreo, entre ellas: *Synthetic Minority Over-sampling Technique* (SMOTE) y *Tomek links*, respectivamente.

Bibliografía

[1] L. E. George and L. Birla, "A study of topic modeling methods," in *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 109–113, June 2018.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[3] J. Yeh, C. Lee, Y. Tan, and L. Yu, "Topic model allocation of conversational dialogue records by latent dirichlet allocation," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*, pp. 1–4, Dec 2014.

[4] Z. Qiu, B. Wu, B. Wang, C. Shi, and L. Yu, "Collapsed gibbs sampling for latent dirichlet allocation on spark," in *Proceedings of the 3rd International Conference on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications - Volume 36*, BIGMINE'14, pp. 17–28, JMLR.org, 2014.

[5] D. Kochedykov, M. Apishev, L. Golitsyn, and K. Vorontsov, "Fast and modular regularized topic modelling," in *2017 21st Conference of Open Innovations Association (FRUCT)*, pp. 182–193, Nov 2017.

[6] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O'Reilly Media, 2009.

[7] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, (Valletta, Malta), pp. 45–50, ELRA, May 2010. `http://is.muni.cz/publication/884893/en`.

[8] M. D. Hoffman, D. M. Blei, and F. Bach, "Online learning for latent dirichlet allocation," in *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, NIPS'10, (USA), pp. 856–864, Curran Associates Inc., 2010.

[9] J. Chuang, C. D. Manning, and J. Heer, "Termite: Visualization techniques for assessing textual topic models," in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '12, (New York, NY, USA), pp. 74–77, ACM, 2012.

[10] C. Sievert and K. Shirley, "LDAvis: A method for visualizing and interpreting topics," in *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, (Baltimore, Maryland, USA), pp. 63–70, Association for Computational Linguistics, June

2014.

[11] M. Röder, A. Both, and A. Hinneburg, "Exploring the space of topic coherence measures," in *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, (New York, NY, USA), pp. 399–408, ACM, 2015.

[12] P. Norvig, "How to write a spelling corrector," February 2007.

[13] C. Whitelaw, B. Hutchinson, G. Y. Chung, and G. Ellis, "Using the web for language independent spellchecking and autocorrection," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, (Stroudsburg, PA, USA), pp. 890–899, Association for Computational Linguistics, 2009.

[14] W. Peter, "The porter stemming algorithm: then and now," vol. 40, pp. 219–223, Jan 2006.

[15] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, (USA), pp. 3111–3119, Curran Associates Inc., 2013.

[16] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, pp. 660–674, May 1991.

[17] Y. Goldberg, *Neural Network Methods for Natural Language Processing*. Canada: Morgan & Claypool Publishers, 2017.