UNIVERSIDAD DE COSTA RICA

SISTEMA DE ESTUDIOS DE POSGRADO

APLICACIONES DE LA MINERÍA DE TEXTO EN LA ENCUESTA NACIONAL DE TRANSPARENCIA 2019: UNA NUEVA ALTERNATIVA DE ANÁLISIS PARA LAS ENCUESTAS DE PERCEPCIÓN

Trabajo final de investigación aplicada sometido a la consideración de la Comisión del Programa de Estudios de Posgrado en Estadística para optar al grado y título de Maestría Profesional en Estadística

JUAN FELIPE GONZÁLEZ ÉVORA

Ciudad Universitaria Rodrigo Facio, Costa Rica

2020

DEDICATORIA

A mi madre y padre, por haber inculcado en mí los valores que me permiten ser la persona que soy hoy. Este documento es dedicado especialmente a ellos: gracias totales.

A mi abuela, que ha sido mi apoyo en todo momento. Gracias infinitas a ella por todo.

A mi compañera, amiga y novia, a quien agradezco por haberme escuchado a lo largo de la construcción del presente documento, y por ser un apoyo incondicional en toda ocasión.

AGRADECIMIENTOS

Agradezco enormemente al profesor Óscar Centeno Mora, quien aceptó ser mi tutor y dedicar una gran cantidad de tiempo en reuniones, revisiones y recomendaciones sobre el análisis y la redacción del documento.

Doy gracias a los profesores Gilbert Brenes Camacho y Johnny Madrigal Pana, quienes aceptaron ser los lectores de este documento.

Gracias a la Universidad de Costa Rica y a la Escuela de Estadística, por la excelencia de sus profesores, con los que compartí y aprendí mucho de lo que sé hoy.

"Este trabajo final de investigación aplicada fue aceptado por la Comisión del Programa de Estudios de Posgrado en Estadística de la Universidad de Costa Rica, como requisito parcial para optar al grado y título de Maestría Profesional en Estadística"

M. Sc. Óscar Centeno Mora

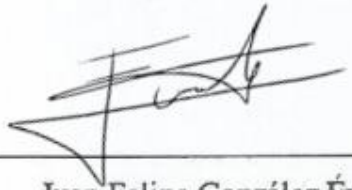
Tutor

PhD. Gilbert Brenes Camacho

Lector

M. Sc. Johnny Madrigal Pana

Lector



Juan Felipe González Évora

Sustentante

Tabla de contenidos

RESUMEN

Codificar y analizar preguntas abiertas provenientes de encuestas de opinión suele ser laborioso y, además, consume tiempo. Para realizar dicho trabajo, interesa poder consignar de forma automática las clases para las preguntas abiertas a partir de la información de las respuestas; asimismo, interesa poder analizar de mejor forma este tipo de información. La minería de texto ofrece una alternativa para ese tipo de problemática. En el presente trabajo, se utilizaron los datos de 12 preguntas abiertas provenientes de la Encuesta Nacional de Percepción sobre la Transparencia del 2019 (ENPT - 2019). Se aplica la minería de texto tanto desde un enfoque descriptivo (frecuencias, redes, clusters y sentimientos) como desde uno predictivo. Este último posee un interés predominante en la presente investigación dado que pretende realizar la codificación automática de respuestas o categorías a partir del aprendizaje automático supervisado. Se emplean algoritmos de máquinas de soporte vectorial, clasificador ingenuo de Bayes, bosques aleatorios, XGBoost y vecinos más cercanos. Los resultados más relevantes muestran que, a partir del análisis descriptivo, se aprecian de mejor forma las descripciones, visualizaciones y relaciones en el análisis de las preguntas abiertas de la ENPT – 2019. El análisis predictivo reseña que los algoritmos con mayor ocurrencia para las preguntas abiertas fueron el clasificador ingenuo de Bayes y los bosques aleatorios, los cuales mostraron precisiones de entre 48% y 76%. Esto permite establecer resultados similares en comparación con los que se obtienen con las categorías que fueron codificadas manualmente. La aplicación de la minería de texto muestra resultados satisfactorios en el análisis integral de las 12 preguntas de la encuesta.

Palabras clave: encuesta de opinión, preguntas abiertas, minería de texto, aprendizaje automático supervisado.

ABSTRACT

Coding and analyzing open-ended questions from opinion surveys is laborious and consume a considerable amount of time. The purpose of this research is to set automatically the classes for open-ended questions based on hand classified examples, in addition to analyze this type of information using sophisticated methods. The text mining applications offer an alternative that facilitates the analysis of the data extracted from the answers in the case of this type of questions in opinion surveys. Twelve open-ended questions from the 2019 National Transparency Perception Survey (ENPT - 2019) were used. Text mining was applied for an exploratory analysis through frequencies, networks, clusters and sentiments. Also, a predictive analysis was performed, having a predominant interest in this research since it intends to perform the automatic coding of responses or categories using supervised machine learning. The algorithms used were Support Vector Machines, the Naive Bayes Classifier, Random Forests, XGBoost and K Nearest Neighbor. The descriptive analysis showed that the representation of text in the analysis of open-ended questions is quite good. The predictive analysis showed that the most selected algorithms with the highest accuracy were the Naive Bayes and Random Forests. The precision of the models selected lies between 48% and 76%. Furthermore, it was shown that the categories predicted by the models chosen for each question allow to establish similar results compared to those obtained with the pre-established categories. The application of text mining showed satisfactory results in the comprehensive analysis of the 12 questions in the analyzed survey.

Keywords: opinion survey, open-ended questions, text mining, supervised machine learning.

Lista de cuadros

Lista de tablas

Lista de gráficos

Lista de figuras

Lista de abreviaturas

Centro de Investigación y Capacitación en Administración Pública de la Universidad de Costa Rica (CICAP-UCR)

CGR: Contraloría General de la República.

ENPT: Encuesta Nacional de Percepción sobre la Transparencia.

ITSP: Índice de Transparencia del Sector Público Costarricense.

XGBoost: Extreme Gradient Boosting.

OOB: Out Of Bagging

UNIVERSIDAD DE COSTA RICA

SEP Sistema de Estudios de Posgrado

Autorización para digitalización y comunicación pública de Trabajos Finales de Graduación del Sistema de Estudios de Posgrado en el Repositorio Institucional de la Universidad de Costa Rica.

Yo, Juan Felipe González Évora, con cédula de identidad 2-0728-0047, en mi condición de autor del TFG titulado Aplicaciones de la minería de texto en la Encuesta Nacional de Transparencia 2019: una alternativa de análisis para las encuestas de percepción.

Autorizo a la Universidad de Costa Rica para digitalizar y hacer divulgación pública de forma gratuita de dicho TFG a través del Repositorio Institucional u otro medio electrónico, para ser puesto a disposición del público según lo que establezca el Sistema de Estudios de Posgrado. **SI** [X] **NO** *[ ]

***En caso de la negativa favor indicar el tiempo de restricción:** _____ **año (s).**

Este Trabajo Final de Graduación será publicado en formato PDF, o en el formato que en el momento se establezca, de tal forma que el acceso al mismo sea libre, con el fin de permitir la consulta e impresión, pero no su modificación.

Manifiesto que mi Trabajo Final de Graduación fue debidamente subido al sistema digital Kerwá y su contenido corresponde al documento original que sirvió para la obtención de mi título, y que su información no infringe ni violenta ningún derecho a terceros. El TFG además cuenta con el visto bueno de mi Director (a) de Tesis o Tutor (a) y cumplió con lo establecido en la revisión del Formato por parte del Sistema de Estudios de Posgrado.

INFORMACIÓN DEL ESTUDIANTE:

Nombre Completo: Juan Felipe González Évora

Número de Carné: B22903 Número de cédula: 2-0728-0047

Correo Electrónico: felipe.2408@hotmail.com

Fecha: 16/04/2020 Número de teléfono: 8343-9944

Nombre del Director (a) de Tesis o Tutor (a): Óscar Centeno Mora

FIRMA ESTUDIANTE

Nota: El presente documento constituye una declaración jurada, cuyos alcances aseguran a la Universidad, que su contenido sea tomado como cierto. Su importancia radica en que permite abreviar procedimientos administrativos, y al mismo tiempo genera una responsabilidad legal para que quien declare contrario a la verdad de lo que manifiesta, puede como consecuencia, enfrentar un proceso penal por delito de perjurio, tipificado en el artículo 318 de nuestro Código Penal. Lo anterior implica que el estudiante se vea forzado a realizar su mayor esfuerzo para que no sólo incluya información veraz en la Licencia de Publicación, sino que también realice diligentemente la gestión de subir el documento correcto en la plataforma digital Kerwá.

CAPÍTULO I: INTRODUCCIÓN

1.1 Contexto

La producción y el crecimiento de datos ha aumentado considerablemente, lo cual representa un reto para el mundo de la analítica (Reinsel, Gantz y Rydning, 2018). En el año 2016, se generaban más de 2,5 quintillones de bytes de datos cada día; de ellos, 80% posee el formato de información no estructurada[1]. Los datos pueden ser comprendidos fácilmente por un humano, pero no por los programas de computadora tradicionales, lo que crea la necesidad de utilizar técnicas para el estudio de grandes volúmenes de información en forma de texto, para un posterior análisis (Eberendu, 2016).

La cantidad masiva de texto contenida en bibliotecas digitales, repositorios, blogs, informes, redes de medios sociales y correos electrónicos produce la necesidad de que las técnicas manuales de la información textual recurran a una forma evolutiva: métodos automáticos, tanto en el procesamiento como en el análisis de este tipo de información. Con el propósito de mecanizar el proceso de análisis de este tipo de información, se han desarrollado técnicas de minería de texto que emplean un conjunto de algoritmos para convertir información textual en datos estructurados, a fin de poder utilizar métodos analíticos en el procesamiento de la información (Maheswari y Sathiaseelan, 2015).

La minería de texto surge ante la necesidad de aportar en el procesamiento de la información textual, así como del interés por agregarle valor. La minería de texto busca aplicar técnicas de análisis de información para brindar significado a los datos no estructurados. Con ello, se persigue convertir este tipo de información para luego generar conocimiento que sea de utilidad tanto para la toma de decisiones como para otros contextos potenciales. (Gurusamy y Kannan, 2014).

[1] La información no estructurada pueden ser datos de texto, imágenes o videos.

El presente trabajo utiliza técnicas de minería de texto aplicadas a preguntas abiertas incluidas en la Encuesta Nacional de Percepción sobre la Transparencia 2019 (ENPT - 2019). El objetivo de esta encuesta es indagar en la ciudadanía la percepción sobre el grado de transparencia en la gestión pública. Por medio de la ENPT – 2019 se intentan conocer las principales dificultades de acceso a la información y participación ciudadana en los temas del manejo de fondos públicos (Contraloría General de la República, 2019). Además, esta encuesta busca determinar la percepción de los ciudadanos y funcionarios públicos respecto del estado del acceso de la información sobre la Hacienda Pública y su relación con la rendición de cuentas y la participación ciudadana sobre su gestión.

El tratamiento que se le da a las preguntas abiertas realizadas en la ENPT - 2019 es subjetivo, laborioso y costoso. Las aplicaciones realizadas en este trabajo muestran una alternativa de análisis distinto para preguntas abiertas en el análisis de texto. Las técnicas utilizadas pretenden mostrar la exploración y la visualización de las respuestas textuales, así como la aplicación de algoritmos que permitan codificar las preguntas abiertas de manera automática.

1.2 Objetivos de la investigación

1.2.1 Objetivo general

- Analizar las preguntas abiertas de la Encuesta Nacional de Percepción sobre la Transparencia 2019 utilizando técnicas de minería de texto, con el propósito de automatizar su procesamiento.

1.2.2 Objetivos específicos

- Realizar un análisis descriptivo de las preguntas abiertas.
- Determinar mediante el análisis de sentimientos el nivel de polaridad en las preguntas abiertas.
- Utilizar técnicas de aprendizaje automático supervisado para crear autocodificadores.

1.3 Justificación

La percepción sobre el desempeño del sector público juega un papel fundamental en la determinación de los niveles de confianza de los ciudadanos hacia el Gobierno. Se suele argumentar que, en muchos casos, las percepciones deficientes del desempeño del sector público se pueden atribuir a la presencia de algún tipo de contradicción de la información entre los ciudadanos y su Gobierno. Al poner a disposición de la ciudadanía más información, las políticas de transparencia pueden lograr que los ciudadanos visualicen y analicen, desde una perspectiva crítica, la manera en que el Gobierno desempeña diversas funciones (Porumbescu e Im, 2015).

El derecho de acceso a la información pública ha sido reconocido a nivel nacional e internacional como un derecho fundamental, por lo que cuenta con un amplio desarrollo jurisprudencial y con un gran impulso desde la perspectiva de los derechos humanos. Si bien el acceso a la información generalmente se asocia con la lucha contra la corrupción y con iniciativas para el mejor manejo de los recursos públicos, es también una herramienta fundamental para que los ciudadanos puedan ejercer el derecho antes mencionado y exigir responsabilidad por parte de los funcionarios públicos. Las leyes de acceso a la información permiten que los individuos y grupos puedan participar de, en primer lugar, las políticas públicas mediante las cuales el Gobierno toma decisiones respecto a proyectos de salud, educación, vivienda e infraestructura; y, en segundo lugar, la discusión acerca de las razones que sustentan tales políticas (Neuman, 2002).

La ENPT - 2019 busca conocer de primera mano la percepción de los ciudadanos y funcionarios públicos acerca del acceso a la información, la rendición de cuentas y la participación ciudadana. Esta iniciativa genera valiosos insumos para la fiscalización superior de la Hacienda Pública y, a la vez, contribuye con la promoción del control ciudadano. Esta encuesta forma parte del Plan Estratégico 2013-2020 de la Contraloría General de la República (CGR), el cual contempla como uno de sus objetivos el incrementar

la transparencia a partir de fomentar el conocimiento ciudadano sobre la administración del sector público, esto con el fin de favorecer el control y la rendición de cuentas (CGR, 2019).

En la ENPT - 2019 se realizaron 12 preguntas abiertas referentes a las percepciones de los ciudadanos y funcionarios públicos sobre el acceso a la información, la rendición de cuentas y la participación ciudadana en el sector público. Se les solicitó a los participantes, de forma abierta, proponer mecanismos de mejora y mencionar las barreras que impiden a las instituciones ser más transparentes en asuntos donde hay fondos públicos de por medio. Las preguntas abiertas son importantes porque no restringen las opciones de respuesta de los encuestados, lo que permite obtener detalles más profundos en las respuestas y, en consecuencia, se puede recabar información valiosa sobre el tema estudiado. Adicionalmente, este tipo de preguntas brinda la oportunidad de que los participantes se expresen libremente, con lo cual se logran conocer detalles importantes que quizá no se tenían contemplados en relación con el tema estudiado.

En adición a lo anterior, las preguntas abiertas no obligan a escoger entre un conjunto fijo de alternativas: son de respuesta libre; por este motivo, según la naturaleza de las preguntas y el interés de la persona, las repuestas varían mucho en cuanto a su extensión y profundidad. El análisis tradicional de este tipo de preguntas consiste en la codificación, que básicamente es el proceso de convertir las respuestas individuales en categorías de manera manual (Rincón, 2014). Luego de la codificación, por lo general, se estiman las frecuencias de las categorías obtenidas, con el fin de que posteriormente sean analizadas por los investigadores involucrados. El proceso de la codificación manual siempre supone una pérdida de información, lo cual implica la existencia de un sesgo asociado a la definición de las categorías concebidas.

La codificación manual de las preguntas abiertas en encuestas es un problema bien conocido de las ciencias sociales; no obstante, hasta ahora solo se tiene conocimiento de unos pocos intentos de automatizar completamente esta tarea. La codificación manual es complicada de estandarizar y es intrínsecamente subjetiva, ya que distintos codificadores

pueden asignar códigos diferentes a los mismos datos de texto, lo que resulta costoso debido a que requiere profesionales especializados para analizar este tipo de datos; además, es una tarea laboriosa y conlleva una cantidad de tiempo considerable. La mayoría de los esfuerzos por simplificar dicha tarea se han dirigido hacia el diseño de sistemas que automaticen la codificación de las preguntas abiertas. Esto puede facilitar, pero no sustituir, el trabajo de los expertos (Giorgetti, Prodanof y Sebastiani, 2002).

El presente trabajo pretende ir más allá en el análisis de las respuestas en preguntas abiertas. Se plantea la utilización de técnicas pertenecientes al campo de la minería de texto, con el fin de mostrar una nueva alternativa en el procesamiento de este tipo de preguntas en las encuestas de percepción. La minería de texto ofrece la posibilidad de descubrir tendencias, patrones, desviaciones y asociaciones de una colección de textos. Con ello se busca, en última instancia, apuntar a descubrir conocimiento en cantidades considerables de información no estructurada (Contreras, 2014).

Visualizar y resumir la información textual mediante técnicas provenientes de la minería de texto se ha convertido en una metodología de análisis de gran utilidad para el procesamiento de las preguntas abiertas. Además, analizar las percepciones de las personas entrevistadas mediante un análisis de sentimientos permite determinar cuáles son las emociones y opiniones relacionadas con la transparencia en el sector público (Kotzé, 2018).

En síntesis, este trabajo muestra una forma de presentar, visualizar y analizar información textual a partir de las preguntas abiertas de una encuesta de percepción. Se utiliza un enfoque en el que la codificación se considera una tarea de clasificación multi-clase, la cual es abordada mediante técnicas desarrolladas originalmente en el campo del aprendizaje automático supervisado. Los algoritmos de dicho campo serán utilizados para codificar automáticamente las preguntas abiertas de la ENPT - 2019.

CAPÍTULO II: ESTADO DE LA CUESTIÓN

Este capítulo aborda tanto el tema de la transparencia en instituciones gubernamentales como el de los fundamentos de la minería de texto. En primera instancia, se expone la conceptualización y presentación de la temática en estudio, con lo cual se muestra la relevancia de la transparencia en las instituciones estatales. Adicionalmente, se explican los componentes que constituyen la transparencia. Luego se presenta una breve descripción de la ENPT - 2019. En segunda instancia, se describen los fundamentos de la minería de texto, del análisis de sentimientos y del aprendizaje automático supervisado en la clasificación textual. Asimismo, se abordan algunos ejemplos de aplicaciones de estos métodos en el abordaje de las preguntas abiertas.

2.1 Transparencia en instituciones gubernamentales

La transparencia es una condición a partir de la cual la información sobre las prioridades, intenciones, capacidades y comportamiento de las organizaciones poderosas está ampliamente disponible para el público global. La trasparencia permite que tanto la acción de la Administración Pública, como el desempeño del Gobierno, sean abiertos y visibles a los ciudadanos. Pese a esto, una mayor transparencia no necesariamente promoverá la democracia y el buen gobierno. (Lord, 2006).

En las últimas dos décadas, los gobiernos de todo el mundo han adoptado leyes de acceso a la información a un ritmo diferente –más acelerado– a cualquier otro momento en la historia. Crear o promover instituciones que fomenten el flujo de información aumenta las posibilidades de inversión y desarrollo económico en una nación; sin embargo, el acceso a la información no necesariamente contribuye a lograr una alta percepción de transparencia en la formulación de políticas de los gobiernos (Relly y Sabharwal, 2009).

Una ventaja de la transparencia radica en que permite a los ciudadanos conocer y opinar sobre el accionar institucional público para diferentes esferas de desempeño. En este sentido, es relevante destacar cómo los mecanismos del poder no son inaccesibles al

escrutinio de los ciudadanos y cómo, a través de los procedimientos institucionales, es factible ahondar en el conocimiento de las oficinas gubernamentales y, de este modo, destacar el trabajo del funcionario público (Aguilera, 2017).

La transparencia se percibe como una fuente de confianza en el Gobierno y, a la vez, como un remedio para la desconfianza del escrutinio ciudadano; en consecuencia, resulta útil medir la transparencia en términos de las percepciones de los ciudadanos. Por otra parte, se ha demostrado que el sentimiento de estar asociado a una política pública específica parece ser más importante para las personas que su participación efectiva en medios electrónicos (Mabillard y Pasquier, 2015).

Una característica relevante de la transparencia es que funciona como una posible forma de luchar contra la corrupción. La libertad al acceso de la información, como característica de una democracia, ayuda a monitorear a los funcionarios públicos, lo que limita sus oportunidades de decantarse por un comportamiento corrupto. Sin embargo, la transparencia en sí misma no es suficiente: el solo hecho de hacer disponible la información a la ciudadanía no evitará la corrupción (Lindstedt y Naurin, 2005).

Otro componente esencial en el que se fundamenta un gobierno democrático es la rendición de cuentas. Por medio de la rendición de cuentas, el Gobierno explica a la sociedad sus acciones y acepta, consecuentemente, su responsabilidad respecto de estas. La rendición de cuentas, según menciona Del Castillo (2003, p.12), es "*el proceso por el cual los funcionarios públicos y gobernantes deben informar y explicar sus decisiones y actos de gobierno, de tal manera que se hagan responsables del ejercicio de la autoridad pública que les es conferida de manera contractual*".

La transparencia como un principio de la administración pública no puede entenderse sin la existencia del ejercicio del derecho de participación ciudadana. Al respecto, Sánchez (2015, p.54) señala que la participación ciudadana es "*el conjunto de procesos mediante los cuales los ciudadanos, a través de los gobiernos o directamente, ejercen influencia en el proceso de toma de decisiones sobre dichas actividades y objetivos*". Por consiguiente, la

participación ciudadana no significa únicamente decidir u opinar, sino también tener la posibilidad de influir en las decisiones que deberán ser tomadas por las instancias de autoridad establecidas en cada caso.

Con base en lo expuesto anteriormente, se puede afirmar que el acceso a la información es importante debido a que, cuanta más y mejor sea la información expuesta para la toma de una decisión en particular, más asertivo será el camino al cual esta guiará. Entonces, el derecho de acceso a la información pública posee una dimensión que responde al plano individual y otra que responde al plano social. Acceder a información no sirve únicamente para satisfacer curiosidades particulares, pues también sirve para crear una conciencia ciudadana en pro de la toma de decisiones colectivas informadas. Esto último permite al Estado contar con la aprobación del pueblo, por un lado, y a los administrados ser fiscalizadores de la gestión de sus gobernantes, por otro lado (Armijo y Vives, 2016).

2.1.1 Acceso a la información en Costa Rica

En Costa Rica, el artículo 27 de la Constitución Política establece que se garantiza la libertad de petición de información, en forma individual o colectiva, ante cualquier funcionario público o entidad oficial, y el derecho a obtener pronta resolución. El artículo 30, por su parte, indica que se garantiza el libre acceso a los departamentos administrativos siempre que existan propósitos de información sobre asuntos de interés público. Asimismo, el artículo 11 señala que la Administración Pública, en sentido amplio, estará sometida a un procedimiento de evaluación de resultados y rendición de cuentas, con la consecuente responsabilidad personal para los funcionarios por el cumplimiento de sus deberes. En todos los casos, la ley señalará los medios para que este control de resultados y rendición de cuentas opere como un sistema que cubra todas las instituciones públicas (Constitución Política de Costa Rica, 2019).

En el año 2017 se emitió el Decreto Ejecutivo Nº 40200 MP-MEIC-MC, el cual señala que el Estado está llamado a efectuar todas aquellas acciones necesarias para resguardar el derecho de acceso a la información pública, entendido como un derecho democrático esencial

para afianzar la gobernanza, el principio de transparencia, la rendición de cuentas y la participación ciudadana. El Gobierno de la República de Costa Rica, por lo tanto, está comprometido a dar los pasos necesarios para fortalecer decisivamente el derecho a la información pública en todas sus manifestaciones, como herramienta indispensable para el logro de una sociedad abierta y transparente.

En el 2003, la resolución 2120-03 de la Sala IV estableció que, bajo el marco del estado social y democrático de derecho, todos y cada uno de los entes y órganos públicos que conforman la administración respectiva deben estar sujetos a los principios constitucionales implícitos de la transparencia y la publicación de la información. Esto debe ser la regla de toda la actuación en la función administrativa (Zamora, 2016).

Existen aún muchas más leyes, decretos y resoluciones que abarcan el tema de transparencia y del acceso de la información en Costa Rica –no se mencionan aquí dado que un tratamiento exhaustivo de estas escapa a los objetivos del presente trabajo–; además, existen distintas instituciones que velan por el cumplimiento de la jurisprudencia respecto a este tema.

2.1.2 Medición de transparencia en Costa Rica

2.1.2.1 Índice de Transparencia del Sector Público Costarricense

En Costa Rica se implementa el Índice de Transparencia del Sector Público Costarricense (ITSP) como un instrumento de medición de transparencia de las instituciones que conforman la Hacienda Pública. Este índice está conformado por cuatro dimensiones: acceso a la información, rendición de cuentas, participación ciudadana y datos abiertos del gobierno. El ITSP se basa en el acceso a la información que las instituciones públicas han colocado en sus sitios web, a fin de enfocar el proceso en la perspectiva del ciudadano. El índice es elaborado por la Defensoría de los Habitantes, en coordinación con el Centro de Investigación y Capacitación en Administración Pública de la Universidad de Costa Rica (CICAP-UCR) y RACSA Gobierno Digital (Zamora, 2016).

El ITPS es desarrollado utilizando una metodología científica y está basado en las mejores prácticas internacionales para medir el estado de la transparencia que, en un momento dado, se puede observar en los sitios web de las instituciones públicas (Zamora, 2016). Sin embargo, el alcance de este índice se limita a medir el acceso de la información de las páginas web de las distintas instituciones, sin tomar en cuenta la opinión de la ciudadanía. Es por esto que la CGR realizó por primera vez en el 2016 la Encuesta Nacional de Percepción de la Transparencia (ENPT), con el fin de conocer la percepción en torno a temas referentes a la transparencia en el sector público desde una perspectiva ciudadana.

2.1.2.2 Encuesta Nacional de Percepción sobre la Transparencia

La ENPT tiene como objetivo conocer la percepción de la ciudadanía en materia de transparencia pública en Costa Rica, esto con la finalidad de apoyar la fiscalización superior de la Hacienda Pública y, a la vez, contribuir con la promoción del control ciudadano y del control político de la Asamblea Legislativa. La ENTP - 2016 abordó tres módulos y entrevistó a un total de 3.297 personas sobre las temáticas de acceso a la información pública, la transparencia para la rendición de cuentas y la participación ciudadana. Para la ENPT - 2019 se agregó un módulo dirigido a funcionarios públicos, el cual es analizado en este trabajo.

La definición de transparencia que se utilizó para esta encuesta es la mencionada por Shuster y Martínez (2014), quienes la definen como "*la apertura de la información de las instituciones u organismos políticos y burocráticos al escrutinio público, mediante sistemas de clasificación y difusión que reducen los costos de acceso a la información del gobierno*". Además, este menciona que la transparencia en el sector público es una expresión y un requisito de los sistemas de gobiernos democráticos, que se plantean como objetivo el sometimiento al escrutinio público de las actividades y resultados del Estado.

Los resultados de la ENTP - 2016 provienen de lo que se conoce como preguntas cerradas. Para la ENPT - 2019 se busca, además, conocer la opinión de las personas entrevistadas utilizando preguntas abiertas. En la siguiente sección se realiza una breve descripción sobre este último tipo de preguntas.

2.1.2.2.1 Preguntas abiertas en encuestas

Las preguntas abiertas permiten a las personas entrevistadas responder libremente con sus propias palabras y brindar los detalles necesarios para aclarar su respuesta (Fuchs y Bošnjak, 2014). La información que se recopila mediante este tipo de preguntas contiene información más detallada y descriptiva, a diferencia de las preguntas cerradas, las cuales arrojan respuestas estrechas y limitadas (Züll, 2016). Además, las preguntas con respuestas libres permiten fijar el nivel jerárquico de detalle según los diferentes niveles de experiencia o información con los que cuenten los encuestados sobre el tema en estudio (Reja, Lozar, Hlebec y Vehovar, 2003).

Entre algunas de las limitaciones de este tipo de preguntas, Aigneren (2008) señala que quedan muchas veces a juicio de la subjetividad: primero, en la transcripción de la respuesta por el entrevistador y, segundo, en su interpretación por parte del codificador y del analista. Adicionalmente, en ciertos casos las preguntas abiertas desconciertan a los sujetos participantes, quienes al no poder responder necesitan de la ayuda del entrevistador. Esto podría originar situaciones de sugestión, lo que genera el riesgo de una distorsión de los resultados (Álvarez, 2003).

En adición a lo anterior, Rincón (2014) menciona que muchas de las decisiones de asignación y de reagrupamiento en la codificación manual se toman sin un análisis global del corpus que considere toda su diversidad, complejidad y riqueza, pues este proceso se realiza en una etapa preliminar en la cual no se ha analizado el archivo con todos los datos. Además, el análisis y procesamiento tradicional de las preguntas abiertas puede requerir mucho tiempo y debe ser realizado por un especialista en la temática de estudio (Fuchs y Bošnjak, 2014).

Estudiar las respuestas provenientes de preguntas abiertas, desde una perspectiva analítica, se refiere a analizar información de tipo textual. Con la llegada y el uso del computador, y con el desarrollo de técnicas estadísticas, dicha labor se ha facilitado y se han superado los inconvenientes antes mencionados. Como otros ejemplos de inconvenientes se podrían mencionar el establecimiento del codificador, al convertir las respuestas individuales

en categorías de manera manual, y el empobrecimiento del contenido de la respuesta abierta, al perderse información (Montenegro y Pardo: 1998, citados por Rincón, 2014). En el presente trabajo, para el análisis de información de tipo textual se utilizarán las técnicas de minería de texto que se describen en el siguiente apartado.

2.2 Fundamentos de la minería de texto

Hearst (1999) se refiere a la palabra minería como la extracción de trozos de mineral de una roca que, de otro modo, no tendrían ningún valor. La minería de datos se refiere a la extracción no trivial de información implícita, previamente desconocida y potencialmente útil que se encuentra en grandes bases de datos. La búsqueda de patrones se lleva a cabo utilizando métodos estadísticos, matemáticos y algorítmicos (Zaïane, 1999). La minería de datos descubre el conocimiento de datos estructurados, mientras que la minería de textos descubre y extrae el conocimiento de datos que no están estructurados, provenientes del lenguaje natural (Ananiadou, Keek y Tsujii, 2006).

Las primeras aplicaciones de la minería de texto nacen en el campo de la recuperación de información, es decir, la actividad de buscar recursos de información (generalmente documentos) de una colección de conjuntos de datos no estructurados que satisfacen las necesidades de la búsqueda. La recuperación de información se centró en facilitar el acceso a esta, en lugar de analizarla, y en encontrar en ella patrones ocultos; este último es el objetivo principal de la minería de texto (Allahyari et al., 2017).

Por su parte, Tufféry (2011) menciona que la minería de texto es el conjunto de técnicas y métodos utilizados para el procesamiento automático de datos de texto, provenientes del lenguaje natural, que se encuentran disponibles en cantidades considerablemente grandes en forma de archivos informáticos. Asimismo, estas técnicas parten del objetivo de extraer y estructurar los contenidos y temas de los datos de texto, con el propósito de obtener información a partir de un análisis rápido (no literario) y descubrir datos ocultos para la toma de decisiones automática. Bajo el mismo esquema, Hearst (2003) define la minería de texto como el descubrimiento por computadora de patrones, previamente

desconocidos, mediante la extracción automática de información de diferentes recursos escritos. Un elemento clave aquí es la vinculación entre la información extraída, a fin de formar nuevos hechos o nuevas hipótesis que se exploran con métodos más convencionales del campo de la Estadística.

El análisis de minería de texto requiere una mezcla entre los saberes y técnicas de la Lingüística y la Estadística. La minería de datos representa la fusión de una serie de otras disciplinas, especialmente la Estadística y el Aprendizaje automático. Muchos de los algoritmos de la minería de datos se basan en estadística y métodos de probabilidad (Tufféry, 2011).

A la minería de datos textuales se le ha llamado procesamiento estadístico de textos, descubrimiento de conocimientos en texto, análisis inteligente de textos o procesamiento de lenguaje natural (Solka, 2018). Entre los mayores temas estudiados en la minería de texto se encuentran la extracción de palabras clave, la clasificación de documentos según un tema en específico y la clasificación de textos debidamente etiquetados (aprendizaje supervisado) (Berry y Kogan, 2010).

Las técnicas de minería de texto existentes se basan en métodos de razonamiento inductivo, a partir de datos extraídos de textos. Este tipo de razonamiento consiste en obtener patrones o modelos generales que expliquen los datos, yendo de lo particular a lo general. Este tipo de técnicas pierde parte de la semántica del texto, debido a que este contiene conocimiento más complejo con el que se puede realizar otro tipo de razonamiento; por lo tanto, el texto proporciona información que las técnicas actuales podrían llegar a omitir (Consuelo, 2017).

Por otra parte, los datos de texto pueden analizarse en diferentes niveles de representación: se pueden tratar como una bolsa de palabras o como una cadena de palabras. En la mayoría de las aplicaciones sería deseable representar la información del texto de forma semántica, con el propósito de se pueda realizar un análisis y una minería de datos que contenga mayor significado. Los métodos modernos de procesamiento del lenguaje natural

aún no son lo suficientemente robustos para funcionar adecuadamente en dominios textuales sin el uso de restricciones para la generación de una semántica precisa del texto. La mayoría de los enfoques de minería de texto en la actualidad todavía dependen de la representación más superficial basada en palabras, especialmente el enfoque de bolsa de palabras (Solka, 2018).

La minería de texto ha resultado de mucha utilidad en una gran cantidad de aplicaciones; sin embargo, se suele ignorar que el texto es un formalismo de representación con una capacidad expresiva mucho mayor que las estructuras de datos. Al reducir el contenido del texto a una forma intermedia, se pierde una gran cantidad de información valiosa para la obtención de nuevo conocimiento. El texto contiene conocimiento, expresado mediante lenguaje natural, mucho más rico que una estructura de datos; por ejemplo, los textos incluyen de manera habitual expresiones condicionales, disyunciones, negaciones, entre otros muchos recursos expresivos (Consuelo, 2017).

Por último, uno de los campos de la minería de texto se ocupa de analizar los sentimientos generados a partir de las respuestas de las personas. En el siguiente apartado, se explica brevemente en qué consiste el análisis de sentimientos, qué se entiende por sentimientos y cuáles son posibles limitaciones que se pueden encontrar al aplicar este método. Se brindan, también, algunos ejemplos de aplicaciones en el caso de preguntas abiertas.

2.2.1 Análisis de sentimientos

El análisis de sentimientos es desarrollado en el año 2001 por Das y Chen[2]. También llamado minería de opinión, extracción de opinión, minería de sentimientos, entre otros, es el campo de estudio que analiza las opiniones, sentimientos, evaluaciones, valoraciones,

[2] Estos autores estaban interesados en extraer los sentimientos de los inversionistas en los tableros de mensajes de la bolsa de valores.

actitudes y emociones de las personas hacia entidades tales como productos, servicios, organizaciones, individuos, problemas, eventos, temas y sus atributos (Liu, 2012).

La palabra sentimiento se entiende como un componente sensorial de una experiencia, el cual se expresa en el estado de ánimo de una persona; o bien, en un sentido más restringido, un sentimiento remite a un estado mental compuesto de elementos afectivos, cognitivos y motivacionales (Rosas, 2010). Los sentimientos pueden ser reconocidos como emociones, juicios, opiniones o ideas, motivados por emociones o susceptibilidad. En la lingüística computacional, la atención se centra en las opiniones y los sentimientos más que en los sentimientos basados en emociones. En este contexto, las palabras sentimiento y opinión se usan a menudo de modo alterno (Banea, Mihalcea y Wiebe, 2011).

Existen dos tipos de enfoques principales para aproximarse al problema de extraer el sentimiento de un texto de forma automática. El primero se basa en un lexicón, lo que implica calcular la orientación de un texto a partir de una serie de palabras o frases que conforman un determinado lecto. El segundo enfoque implica construir clasificadores con textos previamente etiquetados, según la polaridad de su sentimiento (Taboada, Brooke, Tofiloski, Voll y Stede, 2011).

Los léxicos de sentimientos juegan un papel clave en la tarea de análisis de sentimientos. Si el léxico asigna incorrectamente la fuerza del sentimiento o la orientación a las palabras, la precisión del análisis del sentimiento resultante se verá afectada negativamente. La principal limitación del enfoque basado en léxico reside en la calificación incorrecta de las palabras de opinión por parte de los léxicos existentes (Lavillle, Gauch y Alfarhood, 2017). Para afrontar este problema, se introduce vocabulario específico del dominio estudiado, con el fin de mejorar la eficacia de la clasificación de sentimientos.

Respecto de lo anterior, la tarea de clasificar un texto en un sentimiento positivo o negativo se vuelve, en ocasiones, complicada: diferentes personas pueden no ponerse de acuerdo en cuanto a otorgarle una clasificación única; por ejemplo, un mismo texto puede ser interpretado de forma diferente en función de factores culturales, de dominio, de idioma

o, incluso, personales. Por consiguiente, para poder extraer información se debe disponer de un conjunto suficientemente amplio de opiniones y de concesos acerca de dicha definición (Rincón, 2016).

Finalmente, la polaridad de los sentimientos es una característica particular de los textos. La tarea de clasificación binaria, al etiquetar un texto que expresa una opinión, se denomina clasificación de polaridad del sentimiento; en concreto, es la capacidad de determinar si una opinión es positiva o negativa. Más allá de una polaridad básica, también se puede obtener un valor numérico dentro de un rango especificado, de modo tal que, de una determinada forma, se trate de obtener una valoración objetiva asociada a una opinión específica (Pang y Lee, 2008).

2.2.1.1 Algunas aplicaciones de análisis de sentimientos

Entre las primeras aplicaciones en las que se realiza una clasificación de textos según los sentimientos expresados, se encuentra la de Turney (2001). Este estudio presenta un algoritmo de aprendizaje no supervisado, el cual rastrea las discusiones en línea sobre películas. En dicho estudio, se muestra una serie de tiempo con los sentimientos positivos y negativos contenidos en el texto analizado. A partir de estos datos, se generaron dos clasificaciones de películas: recomendadas y no recomendadas.

Por su parte, Madhusudanan, Gurumoorthy y Chakrabarti (2016) realizaron una investigación cuya finalidad era adquirir conocimiento sobre problemas en ensamblaje de aviones; estos problemas que se encontraban expuestos en distintos documentos. Los autores se plantearon encontrar la presencia de problemas detectando sentimientos negativos; sin embargo, los lexicones disponibles para realizar análisis de sentimientos no resultaron suficientes para detectar sentimientos negativos en dominios especializados. Como solución, los investigadores crearon un léxico para identificar problemas específicos del dominio de estudio, el cual se centró especialmente en la identificación de sentimientos negativos.

Otra aplicación es la de Jain (2014), quien utiliza el análisis de sentimientos en preguntas abiertas para estudiar un sondeo, dirigido a estudiantes de secundaria, que tenía como objetivo indagar el interés en cursos o carreras relacionados con la computación y la tecnología. El sondeo se llevó a cabo antes y después de un programa. Las respuestas fueron clasificadas como positivas o negativas, y el análisis resultó de utilidad para la retroalimentación y la mejora en la toma de decisiones del programa.

Por último, Georgiou et al. (2015) se propusieron como objetivo evaluar la dificultad de analizar los sentimientos en el campo de la salud, con el fin de contribuir positivamente a la experiencia tanto del paciente como del médico. Para esto, los autores diseñaron un cuestionario en línea en el que se obtuvieron comentarios de los pacientes acerca de la evaluación de los servicios de atención médica.

2.2.2 Aprendizaje automático en clasificación de textos

Se han llevado a cabo intentos para automatizar la tarea de codificar las preguntas abiertas en las encuestas. Estos esfuerzos se han enfocado en técnicas simples de la recuperación de texto, las cuales buscan hacer coincidir la respuesta que se quiere codificar automáticamente con las descripciones o códigos asignados a la respuesta; o bien buscan detectar la similitud entre ambas (Viechnicki, 1998). En el presente trabajo, se formula el problema de la codificación automatizada en preguntas abiertas de encuestas como un problema de categorización de textos multiclase, el cual se aborda mediante el uso de técnicas de aprendizaje automático supervisado. El aprendizaje supervisado se llama *supervisado* debido a la presencia de la variable objetivo, que sirve para guiar el proceso de aprendizaje (Hastie, Tibshirani y Friedman, 2001).

El aprendizaje automático, *machine learning* en inglés, es un área que estudia cómo construir programas de computadoras o algoritmos que mejoren su desempeño en alguna tarea gracias a la experiencia acumulada en la resolución de esta. Se basa en las ideas de diversas disciplinas, como la Inteligencia Artificial y la Estadística, y en la aplicación de la

probabilidades y complejidad computacional. Para utilizar este abordaje de aprendizaje, se deben considerar una serie de decisiones que incluyen la selección del tipo de entrenamiento, la función objetivo por ser aprendida y el algoritmo necesario para aprender esa función a partir de datos de entrenamiento (Mitchell, 1997).

De acuerdo con Mitchell (1997), los algoritmos de aprendizaje automático han demostrado ser útiles en diversos campos de aplicación, como en la minería de datos, donde han sido aplicados a grandes bases de datos que contienen información implícita, la cual puede ser descubierta en forma automatizada. El principal desafío del proceso de aprendizaje automático consiste en obtener un algoritmo que posea buena capacidad de generalización, es decir, que no solo aprenda a clasificar las observaciones que se utilizaron en el proceso de entrenamiento del modelo, sino que también sea capaz de construir un modelo general que permita clasificar adecuadamente nuevos ejemplos que le son desconocidos (Alfaro, Cárdenas y Olivares, 2014).

En la clasificación automática de textos es necesaria la presencia de un conjunto de categorías $C = \{c_1, ..., c_{|c|}\}$ y un corpus inicial $D = \{d_1, ..., d_{|D|}\}$, el cual contiene una colección de documentos etiquetados con C. A través de un proceso inductivo, el clasificador aprende las características de cada una de las categorías del conjunto de entrenamiento $Dt = \{d_1, ..., d_{|Dt|}\}$. La clasificación de textos puede ser formalizada como la tarea de aprender una función objetivo $F: Dt \rightarrow C$, llamada clasificador (Sebastiani, 2005). En el caso de un análisis de encuestas con preguntas abiertas, el valor de C representa las clases predefinidas y otorgadas a las respuestas de las preguntas abiertas, mientras que D representa cada una de las respuestas.

La codificación automatizada de una encuesta requiere que cierta cantidad de datos se codifique manualmente, a fin de construir el conjunto de datos que funciona para entrenar los algoritmos (Giorgetti, Prodanof y Sebastiani, 2002). La clasificación de documentos textuales es una aplicación de la minería de texto que asigna a los documentos una o más categorías, etiquetas o clases, basadas en el contenido del texto; de ahí que la clasificación

de documentos de texto sea un componente importante de muchas tareas de organización y gestión de la información (Pérez y Cardosa, 2010).

Las limitaciones de las técnicas que se utilizan para la codificación automatizada de preguntas abiertas en encuestas se evidencian en el uso de palabras independientes, puesto que se pierde la estructura semántica de la oración. En un contexto de categorización de texto, lo anterior puede conducir a errores claros. Por ejemplo, podría clasificarse la oración "Estaba enojado con el hombre que insultó a mi esposa" en la categoría de “familia” solo por el hecho de que el clasificador tiende a asociar la palabra "esposa" con la categoría “familia”; sin embargo, difícilmente se clasificaría dentro de tal categoría la oración anterior si se atiende a su significado global (Giorgetti, Prodanof y Sebastiani, 2002).

Existen otro tipo de modelos no supervisados para realizar la clasificación de texto conocidos como *modelado de tema* (*topic modelling* en inglés). El propósito de este tipo de modelos es descubrir patrones en las palabras utilizadas y conectar documentos que compartan características similares (Alghamandi y Alfalqi, 2015). En primera instancia, se debe asumir que existe una cantidad *k* de temas o categorías; en segunda instancia, se estima el modelo y se obtiene una distribución de probabilidad para cada tema sobre cada palabra o unigrama (Liu, Tang, Dong, Yao y Zhou, 2016).

El enfoque que se presenta en este documento es el de aprendizaje supervisado, lo que significa que se debe clasificar previamente el texto.

2.2.2.1 Algunas aplicaciones de la clasificación de textos

Giorgetti y Sebastiani (2003) analizaron tres preguntas abiertas de la Encuesta Social General realizada en Estados Unidos. Un grupo de expertos asignó a cada pregunta un código predefinido. Con estos códigos se entrenaron, y posteriormente compararon, los modelos de máquinas de soporte vectorial y el clasificador ingenuo de Bayes, con el fin de poder predecir las categorías de las respuestas a las preguntas abiertas. Las precisiones obtenidas para dos

de las preguntas oscilaban alrededor del 75%, mientras que para la tercera pregunta la precisión aproximada fue de 37%.

Por otra parte, Dallas y Smith (2015) evaluaron dos técnicas de aprendizaje automático aplicadas al problema de la codificación automatizada de diez preguntas abiertas, esto en la encuesta que analizó las elecciones del 2008 en Estados Unidos. Se comparó la aplicación de dos tipos de métodos de aprendizaje automático para esta tarea: la regresión logística regularizada LASSO y las redes neuronales recurrentes. El estudio encontró que el puntaje para seis de las preguntas resultó ser mayor a 0,85, mientras que para cuatro preguntas el puntaje obtenido osciló entre 0,60 y 0,70.

Por último, Zainuddin y Selamat (2014) utilizaron máquinas de soporte vectorial en un archivo de datos de evaluaciones comparativas, para entrenar a un clasificador de opinión. Los autores analizaron tres preguntas abiertas y compararon los resultados de los modelos utilizando unigramas, bigramas y trigramas. Encontraron que se obtienen mejores resultados con unigramas, puesto que las precisiones aproximadas rondaron el 80%.

En conclusión, las técnicas de minería de texto ofrecen una alternativa de análisis interesante en el abordaje de las preguntas abiertas. En el siguiente capítulo se muestran los materiales, métodos y procedimientos de análisis que se utilizan en este trabajo para el análisis de las preguntas abiertas de la ENPT-2019.

CAPÍTULO IV: METODOLOGÍA

Este capítulo describe los materiales y métodos utilizados tanto en el procesamiento de los datos textuales como en el análisis de las 12 preguntas abiertas de la ENPT - 2019.

4.1 Materiales

A continuación, se describe: la metodología utilizada por la CGR para realizar la ENPT-2019, la población estudiada, el proceso de construcción de los cuestionarios, el tipo de muestreo, la selección de la muestra y, finalmente, las preguntas abiertas de la encuesta (CGR, 2019).

4.1.1 Población y muestreo utilizado

La población de estudio está conformada, por una parte, por los ciudadanos mayores de 18 años residentes en Costa Rica, y, por otra parte, por los funcionarios públicos activos que laboran para instituciones del sector público. La recolección de los datos se llevó a cabo del lunes 11 al viernes 22 de febrero del año 2019. Los datos se obtuvieron en las instalaciones de la CGR utilizando varias herramientas de la plataforma de Google, como el Site, formularios de Google y hojas de cálculo de Google. Estas herramientas permitieron crear cuestionarios en línea para ser aplicados; y estos, a su vez, posibilitaron registrar los resultados en un archivo de datos.

Los datos fueron recolectados mediante una encuesta telefónica. Para la selección de la muestra, se contó con un marco muestral en la consulta a la ciudadanía (números telefónicos celulares)[3] y con un marco muestral en la consulta a los funcionarios públicos[4]. Se utilizó un muestreo simple al azar para seleccionar a los ciudadanos y funcionarios públicos. La estimación del tamaño de la muestra en la ciudadanía se calculó con un nivel de confianza del 95% y con un margen error de 3 puntos porcentuales, lo que determinó un

[3] A partir de un marco muestral de números provenientes de los registros del Instituto Costarricense de Electricidad (ICE).
[4] A partir de la base de datos SICERE, administrada por la Caja Costarricense del Seguro Social (CCSS).

tamaño de muestra mínimo de 1.068 personas por entrevistar. En cuanto a la estimación del tamaño de la muestra de los funcionarios públicos, esta se calculó con un nivel de confianza del 95%, y con un margen de error de 4 puntos porcentuales, lo que determinó que se debía consultar a un mínimo de 600 funcionarios.

4.1.2 Cuestionarios y preguntas abiertas

Se diseñaron tres cuestionarios para la consulta a la ciudadanía y un cuestionario para la consulta a los funcionarios públicos. Cada cuestionario se entendió como un módulo o un componente referido al tema de transparencia. En estos módulos se preguntó acerca de la percepción del acceso a los sitios web de las instituciones públicas, el nivel de participación, y la evaluación de las municipalidades y de las instituciones públicas. Dado que el presente trabajo encuentra su fundamento en el análisis de las preguntas abiertas, estas se presentan en la Tabla 1. Las preguntas se agrupan según el cuestionario en el que aparecen:

Tabla 1. Cuestionarios y preguntas abiertas analizadas

Cuestionario 1: Acceso a la información
P1. Cuando se ha interesado por obtener información sobre las instituciones públicas, ¿qué le ha impedido obtenerla?
P2. ¿Cómo se podría eliminar o solucionar esa barrera?
P3. Desde su propio ámbito, ¿qué podría hacer usted?
Cuestionario 2: Información para la rendición de cuentas
P1. Cuando se ha interesado por obtener información sobre los resultados de lo que hace el sector público, ¿qué le ha impedido hacerlo?
P2. ¿Cómo se podría eliminar o solucionar esa barrera?
P3. Desde su propio ámbito, ¿qué podría hacer usted?
Cuestionario 3: Participación ciudadana
P1. ¿Qué le ha impedido participar de los asuntos del sector público?
P2. ¿Cómo se podría eliminar o solucionar esa barrera?
P3. Desde su propio ámbito, ¿qué podría hacer usted?
Cuestionario 4: Funcionarios públicos
P1. ¿Cuál considera usted que es la principal barrera que le impide a su institución ser más transparente ante la ciudadanía?
P2. Si usted fuera jerarca, ¿qué haría para solucionar o eliminar las barreras que le impiden a la institución ser más transparente ante la ciudadanía?
P3. Desde su propio ámbito, ¿qué podría hacer usted?

Los cuestionarios fueron elaborados internamente en la CGR, mediante el trabajo conjunto del personal de la División de Fiscalización Operativa y Evaluativa, el Despacho Contralor y la División de Contratación Administrativa.

4.1.3 Programa y aplicación web

El programa de análisis de datos empleado fue R Statistics, versión 3.5.2. Se crearon un total de 37 funciones con el fin de preparar los datos, procesarlos y visualizarlos. Para esto fue necesario usar 42 librerías. El código se muestra en el *anexo III.*

También se creó una aplicación web utilizando el programa R Shiny, la cual permite visualizar los resultados obtenidos en este estudio de una manera interactiva. Asimismo, la aplicación web permite, entre otras funciones, cambiar de forma interactiva los parámetros de los distintos análisis y realizar, en tiempo real, predicciones de las respuestas para la pregunta de la ENTP – 2019 relativa a impedimentos para participar en el sector público. Los análisis hechos en este trabajo son mostrados en la aplicación de Shiny.

4.2 Métodos

A continuación, se describe el proceso que requiere el convertir los datos de texto de las preguntas abiertas en información que pueda ser modelada por las técnicas estadísticas tradicionales. En una primera parte, se explica lo relativo a los métodos exploratorios y de análisis de sentimientos. En una segunda parte, se describen los modelos de aprendizaje automático utilizados para predecir las categorías previamente establecidas de las preguntas abiertas. Por último, se explicará la técnica de reducción del número de palabras para entrenar los modelos, así como el proceso de selección y validación de los modelos.

La Figura 1 muestra un esquema del proceso de trabajo en la minería de texto. Primeramente, se efectúa el preprocesamiento del texto, lo que incluye la limpieza de este. Después, se procede a transformar el texto en datos estructurados que puedan ser procesados por las técnicas tradicionales. Luego, se aplica la técnica de análisis de interés. Finalmente, se interpretan de manera sustantiva los resultados obtenidos.

Figura 1. Esquema del flujo de la minería texto

Fuente: Elaboración propia.

4.2.1 Preprocesamiento del texto

Los datos de texto no están estructurados, por lo que, para poder aplicar técnicas de minería de datos, se debe conseguir una representación adecuada que permita realizar los análisis propuestos. Lo primero que se define es el *corpus*, esto es, el documento que contiene texto. En el caso del presente trabajo, el corpus son las respuestas a las preguntas abiertas de la ENTP - 2019. Aunque las preguntas que forman un texto por sí solas no aportan mucha información, pueden facilitar la realización de un primer análisis del corpus textual, ya que es posible determinar ciertos factores no semánticos de este. Para representar los datos textuales se utilizó el método de la bolsa de palabras[5].

La bolsa de palabras es un tipo de representación de los documentos que proviene del área de la recuperación de información. La bolsa toma todas las palabras del documento y elimina sus relaciones sintácticas y semánticas, debido a que se ignora el orden de las palabras. Este modelo solo identifica si las palabras aparecen en el documento; de este modo, se pueden realizar los análisis tradicionales del texto (Consuelo, 2017). En el modelo de bolsa de palabras, cada documento d_i puede representarse por una lista de pares (s_j, f_j), donde s_j es una palabra y f_j la frecuencia de s_j que aparece en d_i (Deng, Li y Weng, 2018).

Los pasos a los que se somete el corpus original, para que pueda ser procesado y analizado, son los siguientes:

[5] En inglés, *bag of words*

1. Corrección de ortografía. Para realizar esta tarea, se creó una función que utiliza el corrector ortográfico *Hunspell*. Esta función remplaza las palabras mal escritas o con faltas de ortografía por las palabras sugeridas por dicho corrector. *Hunspell* es al mismo tiempo una biblioteca y un programa que funciona como corrector ortográfico y analizador morfológico. Esta librería realiza la corrección ortográfica de softwares como LibreOffice, OpenOffice, Mozilla Firefox, Thunderbird y Google Chrome.

2. Tokenización. Un token es una unidad significativa de texto que se desea analizar; por ejemplo, una palabra. La tokenización consiste en dividir el texto en tokens o unidades. En la minería de texto, el token que se almacena en cada columna o fila suele ser una sola palabra, pero también puede ser un N-grama, una oración o un párrafo (Silge y Robinson, 2019).

3. Remover palabras vacías (en inglés, *stop words*). Las palabras vacías son aquellas que aparecen frecuentemente en el texto, pero no aportan significado relevante; por ejemplo, los artículos, las preposiciones y las conjunciones. En español, las siguientes podrían ser consideradas palabras vacías: aún, este, ese, están, aquel y para, entre otras.

4. Remover los símbolos. Todos los símbolos, así como cualquier elemento extraño, es eliminado del corpus.

5. Remover números. Los números fueron eliminados del *corpus* debido a que no brindan información útil para el análisis.

6. Remover espacios extra en blanco. Los espacios vacíos extra son eliminados ya que pueden afectar el proceso de codificación de las palabras del corpus.

7. Conversión a minúsculas. Todas las palabras en el texto son convertidas a minúsculas. Esto ayuda a reducir el número de palabras, al no tomar en cuenta elementos distintos.

8. Lematización (en inglés, *stemming*). Este es un proceso de normalización de las palabras que consiste en transformarlas en una forma uniforme: su raíz. Tal proceso es relevante debido a que, para ciertos análisis, se puede reconocer cuándo las palabras presentan un

significado similar. Otra ventaja consiste en que la lematización reduce el tamaño del vocabulario. Un ejemplo de lematizar un texto es el siguiente: texto original: “La falta de información y la burocratización”; texto con lematización: “La faltar de información y la burocracia”.

4.2.2 Análisis exploratorio

El análisis exploratorio de los datos textuales utiliza las técnicas de nubes de palabras, correlaciones, análisis de redes y cluster jerárquico. Una nube de palabras es un término que se usa para denominar una representación visual de las palabras más frecuentes en un texto. El uso de esta técnica puede ofrecer un primer acercamiento a la representación de los temas principales presentes en las respuestas a preguntas abiertas. En una nube, el tamaño de las palabras es mayor para las que aparecen con más frecuencia y, por consiguiente, menor para las que aparecen con menos frecuencia.

La nube de palabras se puede generar particionando el texto en una sola palabra, en dos o en más de dos. La tokenización se puede realizar mediante N-gramas. Esta es una subsecuencia de n elementos de una secuencia de texto dada (Milios et al., 2007). Si N=2, las subsecuencias se denominan bigramas; si N=3, trigramas; si N >= 4, N-gramas. Estos permiten estudiar los textos con un mejor entendimiento, debido a que es un método en el que se pueden apreciar mejor las respuestas, según su estructura semántica.

Para el análisis descriptivo de texto también se puede emplear el análisis de redes de palabras. Este permite apreciar las palabras con mayor centralidad. Asimismo, en la red de palabras se suelen calcular indicadores, los cuales aportan medidas de centralidad según los nodos. Los indicadores más comunes suelen ser:

Grado total: el grado de una palabra en una red es el número de conexiones que mantiene con otras palabras.

Grado interno: se refiere al número de conexiones que recibe una palabra, es decir, el número de veces que la palabra aparece después de otra.

Grado externo: se refiere al número de conexiones que salen de una palabra, o sea, el número de veces que la palabra aparece antes de otra.

Otra alternativa de análisis en la red de palabras consiste en determinar el grado de asociación entre estas. Para esto, se calcula el coeficiente de correlación de Mathews, también conocido como el coeficiente phi. Este devuelve un valor entre 0 y 1. Una correlación de 1 indica que las dos palabras aparecen juntas en todas las respuestas, mientras que un valor de 0 significa que las palabras nunca aparecen en la misma respuesta. El coeficiente phi para la palabra X y la palabra Y está dado por la siguiente ecuación:

$$\Phi = \frac{n_{11}n_{00} - n_{10}n_{01}}{\sqrt{n_1 n_0 n_{0*} n_{1*}}}$$

En esta ecuación se representa lo siguiente: n_{11}, el número de respuestas donde aparecen tanto la palabra X como la palabra Y, n_{10} y n_{01}, los casos en que una palabra aparece sin la otra, n_0, el número donde no aparece la palabra X, n_1, el número donde aparece X, n_{0*}, el número donde no aparece la palabra Y, y n_{1*}, el número donde aparece Y (Silge y Robinson, 2019).

Utilizando el mismo esquema del análisis exploratorio, se analizan las relaciones jerárquicas entre las palabras mediante un análisis de cluster jerárquico. En este análisis, las palabras son representadas por medio de un dendrograma. El método utilizado para unir los grupos fue el de Ward. Este método indica la distancia entre dos grupos, A y B, y está determinado por la siguiente ecuación (Murtagh y Legendre, 2011):

$$\Delta(A,B) = \frac{n_A n_B}{n_A + n_B} ||m_A - m_B||^2$$

En esta ecuación, m es el centroide para cada grupo y n representa la cantidad de casos en cada grupo.

Para finalizar, otra forma de realizar el análisis de conglomerados es mediante un árbol filogenético. Un árbol filogenético es un esquema arborescente que muestra las relaciones entre varias entidades que se cree que poseen características en común. Esta

estructura de presentar la información proviene de la biología; sin embargo, se puede aplicar para presentar datos de texto (Vani, Appa, Sridhar, Chakravarthy, Nageshwararo y Rao, 2010).

4.2.3 Análisis de sentimientos

El enfoque utilizado en este trabajo es el del lexicón, el cual pertenece al campo de los algoritmos de aprendizaje no supervisado. En dicho enfoque, se utiliza un diccionario o un lexicón, el cual se compone de una lista de palabras asociadas que presentan una polaridad específica y un puntaje según la fuerza de la polaridad (Labille, Gauch, y Alfarhood, 2017). Existen diversos diccionarios para realizar la tarea del análisis de sentimientos en español, entre ellos LIWC (Linguistic Inquiry and Word Count) y Sentitex; sin embargo, estos no proporcionan un ajuste exacto al contexto de la presente investigación, debido al dominio de estudio tratado: la transparencia.

El análisis de sentimientos es sensible al contexto o tema en el que se aplica, dado que las palabras pueden remitir a un sentimiento o a un significado diferente en distintos dominios. En el tema de la transparencia no existen diccionarios para realizar un análisis de sentimientos, por esto fue necesario crear un lexicón referente al tema. De los cuatro cuestionarios que contiene la ENTP - 2019, se estudió la pregunta que hace referencia a las barreras. Las preguntas estudiadas apuntan en un mismo sentido, lo que permite crear un único diccionario para analizarlas. Las demás preguntas no fueron consideradas para realizar este análisis debido a que se habría debido crear un diccionario particular para cada una.

El estudio aquí presentado se centró en identificar los sentimientos u opiniones negativas. Para llevar a cabo esta tarea, se creó un diccionario manual. Se pueden distinguir dos etapas en la creación de léxicos: en primer lugar, la generación de la lista de palabras que contienen sentimientos; en segundo lugar, la asignación de la polaridad de las palabras (Kotelnikov, Bushmeleva, Razova, Peskisheva y Pletneva, 2016).

Con el propósito de crear la lista de palabras, se generaron frecuencias de las palabras presentes de las distintas preguntas de la ENTP - 2019; de esta manera, se extrajeron las

palabras que mostraban una connotación negativa en el tema de barreras e impedimentos a la hora de participar u obtener información del sector público. El lexicón creado en este trabajo contiene un total de 239 palabras, y se puede apreciar en el Cuadro 16 que se encuentra en el *anexo II*.

Una vez identificadas las palabras, se procedió a asignar a cada una un puntaje según la fuerza de la polaridad. Para asignar los puntajes, se realizó una evaluación interna en la que se discutió qué acciones reflejaban una mayor negatividad. A manera de ejemplo, en el caso de que las instituciones públicas nieguen la información que se les solicita, se esperaría que tal negativa repercuta en un peso mayor en el puntaje de negatividad; a esta situación podrían seguir, en orden decreciente según dicho puntaje, el que la información no se encuentre disponible en las páginas web o que esté desactualizada, los temas relacionados con la burocracia, y el acceso a la información. Con base en estos criterios, se asignaron los puntajes considerando una escala de 1 a 6, en la que 1 indica que la palabra posee una menor negatividad y 6 una mayor negatividad.

Por otra parte, una de las tareas más importantes en el análisis de sentimientos es determinar la secuencia de palabras afectadas por la negación. Pang y Lee (2002) proponen añadir la etiqueta ‘no_’ a las palabras que aparecen seguidas de las negaciones ‘no’ y ‘nunca’. A modo de ejemplo, las oraciones ‘es bueno’ y ‘no es bueno’ indican orientaciones de sentimientos opuestas. Si se implementa este método en la segunda oración y se eliminan las palabras vacías, se obtendría ‘no no_bueno’; de esta manera, el algoritmo determinaría la oración como negativa.

Una vez que se dispone del diccionario listo con las palabras y los puntajes, se procede a ejecutar el algoritmo para asignar un puntaje a cada respuesta. En el caso de que la palabra aparezca en el texto, se le asigna el puntaje preestablecido. La asignación de los puntajes posibilita estudiar la distribución de sentimiento en cada una de las preguntas y, además, permite conocer cuáles son las palabras más frecuentes consideradas como negativas.

4.2.4 Clasificación de texto

Una vez que el *corpus* está completamente limpio, se procede a transformar el texto no estructurado para que pueda ser procesado por los algoritmos propuestos. Para esto se crea una matriz documento-palabra [6] en la que cada fila corresponde a una observación o documento (respuesta). Las columnas, por su parte, representan las palabras contenidas en todas las respuestas. Dentro de esta matriz, se encuentra la frecuencia de las palabras que se mencionan en los documentos. En concordancia con el parámetro establecido, en el presente trabajo se eliminaron las palabras que aparecen una única vez en el corpus.

Por otra parte, el problema de clasificación consiste en que, dado un archivo de datos de registros u observaciones (individuos) y un conjunto de clases, se debe encontrar una función tal que cada registro sea asignado a una clase. En este caso, los registros son las respuestas a las preguntas abiertas, las variables predictoras son las palabras que el documento contiene y las clases[7] son las que fueron asignadas manualmente por la persona codificadora. Antes de entrenar los modelos, se deben seleccionar las palabras que se incluirán en estos; para ello, se deben elegir las palabras que estén asociadas con las clases. Este proceso se describe a continuación.

4.2.4.1 Selección de palabras

Un desafío que se presenta al realizar una categorización de textos es la alta dimensionalidad del espacio de las variables. Un documento, generalmente, contiene cientos o miles de palabras distintas, las cuales son consideradas como variables; sin embargo, muchas de ellas pueden ser ruidosas, menos informativas o redundantes con respecto a su clase. Lo anterior puede confundir a los clasificadores y afectar su rendimiento. La selección de palabras debe aplicarse para eliminar este tipo de variables, a fin de reducir el espacio

[6] En inglés como *Document-Term Matrix*
[7] Las técnicas utilizadas requieren del supuesto que las codificaciones manuales sean establecidas correctamente.

dimensional a un nivel manejable; esto mejora la eficiencia y precisión de los clasificadores utilizados (Mladenić, 2011).

Al respecto, Mladenić (2011) señala que la mayoría de los métodos para la selección de palabras que se utilizan en clasificación de texto son muy simples, en comparación con los métodos para seleccionar variables desarrollados en el aprendizaje automático supervisado. En clasificación de texto, se realiza la selección de variables asumiendo la independencia de las palabras, de modo que se asigna una puntuación a cada palabra de forma independiente y, entonces, se seleccionan las palabras que muestran las puntuaciones más altas. No obstante, existen métodos más sofisticados para la selección de variables en datos de texto, los cuales toman en cuenta las interacciones entre las palabras. Por otro lado, los enfoques para la selección de palabras que buscan un espacio de todos los posibles subconjuntos de palabras pueden requerir bastante tiempo cuando se trata de un gran número de palabras; por eso, rara vez se utilizan en datos de texto.

La selección de variables permite lograr un modelo parsimonioso sin que se vea afectada la precisión. La prueba que se utilizó en este caso es la Chi-cuadrado, un método popular en la selección de palabras. Esta prueba estima individualmente para cada palabra el estadístico Chi-cuadrado con respecto a las clases, por lo que se analiza la dependencia entre la palabra y las clases (Zareapoor y Seeja, 2015). El estadístico Chi-cuadrado esta dado por:

$$\chi^2_{(gl)} = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$$

En esta fórmula, O_i son las frecuencias observadas en la tabla, E_i son las frecuencias esperadas, gl son los grados de libertad y k es el número de entradas de la tabla.

Al realizar las pruebas de independencia entre las clases y las palabras, se encontró que la estimación del estadístico Chi-cuadrado podría ser incorrecto debido a que muchos de los valores esperados eran menores que 5, lo que desestabiliza la prueba. Para solucionar el problema, se efectuó una simulación de Monte Carlo, con el fin de estimar la probabilidad

asociada al estadístico Chi-cuadrado. Para cada prueba se hicieron un total de 20.000 Bootstrap.

El criterio utilizado para remover las palabras fue la probabilidad asociada a la chi-cuadrado. Para esto, se creó una función que realiza las pruebas de hipótesis para las variables predictoras y elimina las palabras con una probabilidad dada que son recibidas por la función como parámetro. Para comprobar que al eliminar las variables no se reduce la precisión, se entrenó un modelo de bosques aleatorios con todos los predictores y otro únicamente con las variables seleccionadas. Luego, se comparó la precisión mediante las muestras que quedan por fuera en la agregación de Bootstrap (*Out Of Bagging; OOB*). Una estimación de error mediante OOB es casi idéntica a la obtenida por la validación cruzada con N grupos. A diferencia de muchos otros estimadores, los bosques aleatorios pueden entrenarse en una secuencia, y la validación cruzada se realiza al mismo tiempo (Hastie et al., 2001). Se probó para cada pregunta distintos puntos de corte. Se eligió el que mantuviera la precisión igual o, inclusive, la aumentara, en comparación con entrenar un modelo con todas las variables.

4.2.4.2 Métodos de clasificación

Los modelos que se comparan con el propósito de obtener el mejor clasificador para cada pregunta son el clasificador ingenuo de Bayes, bosques aleatorios, XGBoost, máquinas de soporte vectorial y vecinos más cercanos. Estos se describen en los siguientes subapartados:

La terminología utilizada es la siguiente:

Sea D_i el conjunto de respuestas del corpus formado por $\{d_1, d_2, \ldots, d_n\}$ y C las clases predefinidas por el codificar dadas por $\{c_1, c_2, \ldots, c_j\}$.

4.2.4.2.1 Clasificador ingenuo de Bayes

El clasificador ingenuo de Bayes modela, en cada clase, la distribución de documentos, o, en el caso del presente trabajo, de las respuestas a las preguntas abiertas. Para

ello, este clasificador usa un modelo probabilístico, el cual asume que la distribución de las diferentes palabras es independiente entre sí (Allahyari et al., 2017). En este modelo se captan las frecuencias de las palabras que se encuentran en un documento, esto debido a que el texto se representa como una bolsa de palabras. Los documentos de cada clase se pueden modelar como muestras que provienen de una distribución multinomial de palabras. Como resultado, la probabilidad condicional de que un documento dado corresponda a una clase es simplemente un producto de la probabilidad de cada palabra observada en la clase correspondiente (Aggarwal y Zhai, 2012).

El clasificador ingenuo de Bayes se construye utilizando un conjunto de datos de entrenamiento, esto con el fin de estimar la probabilidad de cada clase dados los valores de las palabras de los documentos. Para esto se usa el Teorema de Bayes, el cual estima las probabilidades (Araujon, 2009):

$$P(c_j | \, d) = \frac{P(c_j)\, P(d \mid c_j)}{P(d)}$$

El denominador de la ecuación anterior indica la probabilidad de obtener el documento, y no distingue entre las clases. Por esta razón, el denominador va a ser igual para cada clase y no necesita ser maximizado; en consecuencia, se puede eliminar de la ecuación. Este método asume que las palabras son condicionalmente independientes, dada la clase. Esto simplifica los cálculos. La ecuación del clasificador de Bayes es dada por:

$$P(c_j | \, d) = \; argmax_{c_j eC}\, P(c_j) \; \prod_{i=1}^{M} P(d_i \mid c_j)$$

A pesar de que el supuesto de independencia condicional es generalmente falso para la aparición de una palabra en documentos, el clasificador ingenuo de Bayes es efectivo (Araujon, 2009).

4.2.4.2.2 Bosques aleatorios

Los bosques aleatorios son una combinación de árboles de decisión. En los bosques aleatorios, un árbol no está correlacionado con ningún otro y los árboles son independientes entre sí (Breiman, 2001). Antes de explicar cómo funcionan los bosques aleatorios, se describen brevemente los conceptos básicos requeridos para poder generar un árbol de decisión.

- Árboles de decisión

Los árboles de decisión se pueden representar en forma completa mediante un gráfico en el que cada partición es binaria, lo que posibilita que su visualización sea sencilla (Hastie et al., 2001). Para generar este modelo, se utiliza el algoritmo de Hunt, el cual se usa para particionar los datos. Este método hace crecer el árbol en una forma recursiva, con el fin de obtener subconjuntos de datos puros. Si un subconjunto es puro, indica que se logró discriminar satisfactoriamente una clase específica.

Sea S_t el conjunto de entrenamiento asociado con el nodo t, y sea $C_j = \{c_1, c_2, \ldots, c_j\}$ el conjunto de etiquetas de las clases. La siguiente es la definición recursiva del algoritmo de Hunt (Kareem y Duaimi, 2014):

1. Si todos los registros en S_t pertenecen a la clase c_j, entonces t es un nodo hoja que se etiqueta como y_j.

2. Si S_t contiene registros que pertenecen a más de una clase, se escoge una variable para dividir los datos en subconjuntos más pequeños. Este algoritmo es aplicado a cada nodo hijo de forma recursiva.

Los árboles de decisión utilizan el método CART[8], el cual es una técnica no paramétrica de segmentación binaria en la que el árbol es construido dividiendo

[8] En inglés, *Classification and Regression Tree.*

repetidamente los datos. En cada división los datos son partidos en dos grupos mutuamente excluyentes. El nodo inicial es llamado nodo raíz y se divide en nodos hijos; luego, el procedimiento de partición es aplicado a cada nodo hijo por separado. Las divisiones se seleccionan de modo que la "impureza" de los hijos sea menor que la del nodo padre. Estas divisiones están definidas por un valor de una variable explicativa (Deconinck et al., 2006, citado por Serna, 2009).

El objetivo del proceso descrito es particionar la respuesta en grupos homogéneos y, a la vez, mantener el árbol razonablemente pequeño. Para dividir los datos se requiere un criterio de aprisionamiento, el cual determinará la medida de impureza. Esta última establecerá el grado de homogeneidad entre los grupos. Para medir la impureza de los nodos, se utilizó el índice de Gini. Este índice esta dado por (Mitchell, 1997):

$$Gini(t) = 1 - \sum_{j} (P(j|t))^2$$

Donde P(j/t) es igual a la probabilidad de pertenecer a la clase j, estando en el nodo t.

Dado el índice de Gini como una medida de la impureza, se procede a estimar la efectividad con la que una variable logra clasificar los datos de entrenamiento. La medida es llamada *ganancia de información*. La variable que presente el mayor valor es la seleccionada para dividir los datos:

$$GananciaDeInformación(t) = Gini(NodoPadre) - \sum_{j} \frac{n}{nj}\, Gini(NodoHijo)$$

Donde n es el tamaño de muestra y n_j el número de individuos en la categoría j.

- Bosques aleatorios

Los bosques aleatorios utilizan el método de agregación de Bootstrap[9], que es una técnica cuya finalidad es reducir la variancia en una función de predicción. Este método

[9] En inglés, *bagging.*

funciona bien en algoritmos con una alta variancia y con sesgo bajo, como es el caso de los árboles de decisión. Los bosques aleatorios utilizan la agregación de Bootstrap, en los que se construyen una gran cantidad de árboles y luego se promedian sus resultados. En el caso de clasificación, se asigna la categoría con la mayoría de los votos otorgados por los árboles individuales.

El componente aleatorio de este algoritmo que reduce la incertidumbre se logra por medio de dos formas. Primero, cada árbol de decisión es entrenado con una muestra independiente de Bootstrap del total de datos de entrenamiento. Segundo, para entrenar cada árbol, se utiliza solamente un subconjunto seleccionado de forma aleatoria de m variables del total de p variables (Cutler, Cluter, y Stevens, 2011). Para el caso de clasificación, el valor predeterminado del número de variables por seleccionar es $\sqrt{p}$ (Hastie et al., 2001).

Por tanto, si se tienen K árboles, la clase estimada estaría dada por:

$$\widehat{y_i} = \sum_{k=1}^{K} f_k(x_i)\,, f_k \in F$$

Donde f_k son los árboles de clasificación individuales y F es la función que contiene todos los árboles.

El hiperparámetro por calibrar en este algoritmo es el número de árboles de clasificación por entrenar.

4.2.4.2.3 XGBoost (*Extreme Gradient Boosting*)

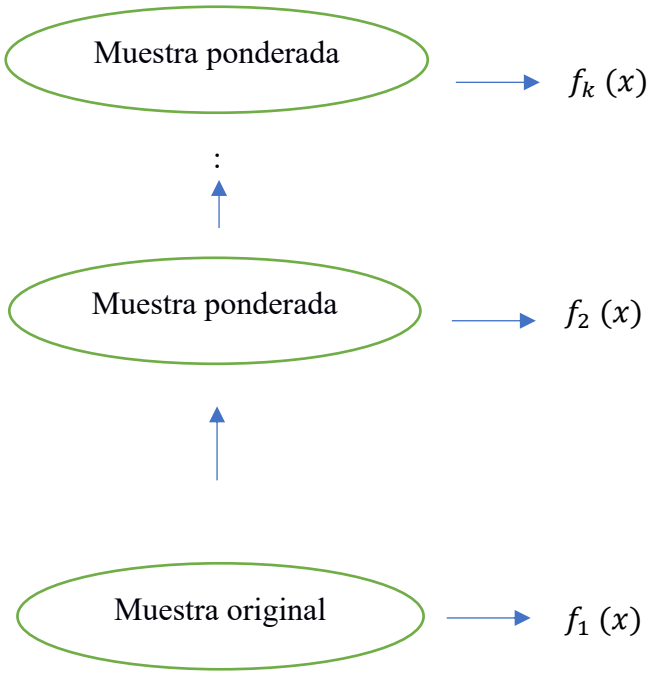
Este algoritmo está conformado por árboles de clasificación, al igual que los bosques aleatorios; sin embargo, funciona de manera distinta. Este modelo, en vez de implementar la agregación de Bootstrap, utiliza el método de potenciación[10]. La potenciación consiste en combinar varios árboles de clasificación débiles para producir un clasificador robusto. Un clasificador débil se caracteriza por brindar resultados que muestran una correlación baja con

[10] En inglés, *boosting*.

las verdaderas clases. Por otra parte, un clasificador robusto muestra un mejor rendimiento que un clasificador débil (Hastie et al., 2001).

El propósito de la potenciación es aplicar secuencialmente el algoritmo de clasificación débil a versiones modificadas del archivo de datos, de manera repetitiva; esto produce una secuencia de clasificadores débiles $f_k\ (x), k\ =\ 1, 2, \ldots, K$. De esta manera, los clasificadores son entrenados en versiones ponderadas del archivo de datos, como muestra a continuación la Figura 2 (Hastie et al., 2001):

Figura 2. Esquema de XGBoost



Fuente: tomado del libro *The Elements of Statistical Learning.*

Las predicciones de todos los clasificadores $f_k(x)$ se combinan según la cantidad mayoritaria de votos ponderados para producir la predicción final:

$$f(x) = \sum_{k=1}^{K} a_k f_k(x)$$

Donde $a_1, a_2, \ldots, a_k$ se calculan mediante el algoritmo de potenciación, el cual se encarga de ponderar la contribución de cada $f_k(x)$. Su efecto es incrementar la influencia de los clasificadores con una mayor precisión (Hastie et al., 2001).

Las modificaciones de la muestra, en cada paso de un clasificador a otro de la potenciación, consisten en aplicar ponderaciones $w_1, w_2, \ldots, w_n$ a cada una de las observaciones de entrenamiento. Inicialmente, todos los pesos se establecen como $w_i = 1/N$, de modo que el primer paso simplemente entrena al clasificador sobre los datos de manera habitual. Para cada iteración sucesiva de k = 2, 3, …, K, los pesos de observación se modifican individualmente y el algoritmo de clasificación se vuelve a aplicar a las observaciones ponderadas. En el paso K, las observaciones que en el paso anterior presentan valores residuales mayores con el clasificador $f_{k-1}(x)$ tendrán sus pesos incrementados, mientras que los pesos se reducen para aquellas observaciones con residuales menores. Así, a medida que avanzan las iteraciones, las observaciones que son difíciles de clasificar correctamente reciben una influencia cada vez mayor (Hastie et al., 2001).

Dado los árboles de clasificación f_k, se define la función objetivo por optimizar para un conjunto de parámetros θ como (Mateo, 2014):

$$Obj(\theta) = \sum_{i}^{n} l(y_i, \hat{y_i}) + \sum_{k=1}^{K} \Omega(f_k)$$

Donde $l(y_i, \hat{y_i})$ representa la función de perdida que compara el valor verdadero con el predicho; en este caso se utilizó el error de clasificación. El segundo término $\Omega(f_k)$ es el término de regularización que penaliza la complejidad del modelo utilizado para evitar sobreajuste[11]. En XGBoost la fórmula de regularización está dada por:

$$\Omega(f) = \Upsilon^T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

[11] El sobreajuste es el efecto de sobreentrenar un algoritmo de aprendizaje con unos ciertos datos para los que se conoce el resultado deseado.

Donde T es el número de nodos j, mientas que el segundo término representa la regulación L2 para los puntajes w_j en cada nodo.

Los hiperparámetros por optimizar en este algoritmo son:

1. Número de árboles: representa la cantidad de árboles por entrenar. Un número grande podría generar sobreajuste en los datos.
2. Eta: controla la tasa en el que el modelo aprende. Valores positivos pequeños de *eta* necesitan de una cantidad de árboles mayor, y viceversa. Este hiperparámetro reduce las ponderaciones de las variables, con el propósito de que el proceso de potenciación sea más conservador.

4.2.4.2.4 k vecinos más cercanos

Este es uno de los métodos más clásicos y simples para enfrentar problemas de clasificación. Este algoritmo clasifica cada observación según la clase mayoritaria presente entre los vecinos más cercanos que se encuentran en el conjunto de entrenamiento. Esto se realiza mediante una función de distancia o similitud. La calidad de la clasificación del método depende de la manera en que se calculan las distancias entre las diferentes observaciones (Nguyen, Rivero y Morell, 2015).

La distancia más común, y la que se utilizó en este trabajo, para entrenar este algoritmo es la Euclidiana (Hastie et al., 2001):

$$Euclidiana(X,Y) = \sqrt{\sum_{i=1}^{n}(xi - yi)^2}$$

Donde X = (x_1, x_2, …, x_m) y Y = (y_1, y_2, …, y_m) representan los valores de las m variables en el archivo de entrenamiento.

En este método se debe elegir el valor de k que optimice el error de clasificación. El valor máximo asignado que puede tomar el valor de k es de $\sqrt{n}$. El algoritmo toma el valor de k que minimice el error de clasificación.

4.2.4.2.5 Máquinas de soporte vectorial

Las máquinas de soporte vectorial se fundamentan en modalidades de hiperplanos para la división de las observaciones. Este método fue creado originalmente para resolver problemas de clasificación lineal; sin embargo, este algoritmo actualmente cuenta con extensiones para ser utilizado en solucionar problemas no lineales. Si los ejemplos no son separables linealmente en el espacio original, la búsqueda del hiperplano de separación –normalmente de muy alta dimensión– se lleva a cabo de forma implícita, utilizando distintos núcleos con el fin de proyectar los datos en un espacio dimensional superior que sea linealmente separable. La idea de tal procedimiento es seleccionar un hiperplano de separación que equidista de los ejemplos más cercanos de cada clase, para obtener lo que se denomina un margen máximo a cada lado del hiperplano. Además, a la hora de definir el hiperplano, solo se consideran los ejemplos de entrenamiento de cada clase que caen justo en la frontera de dichos márgenes. Estos ejemplos reciben el nombre de *vectores soporte* (Carmona, 2014).

Este algoritmo encuentra el hiperplano ideal entre todos los hiperplanos posibles, esto con la finalidad de separar los datos según su categoría. El hiperplano que separa los datos de una mejor manera es el que logra maximizar el margen entre los vectores de soporte. Las funciones de núcleo que son más comunes son la lineal, la base radial y la función sigmoidea (Godoy, 2015). Los tres núcleos fueron probados para calibrar el modelo y encontrar el que mejor ajustara los datos.

Las máquinas de soporte vectorial estándar son propuestas originalmente para resolver problemas de clasificación binaria. En el caso de clasificación multiclase, el problema se resuelve comúnmente mediante una descomposición en varios problemas

binarios, para los cuales el algoritmo de máquinas de soporte vectorial estándar puede ser usado. Para esto se utiliza el enfoque uno contra uno, el cual consiste en entrenar un modelo de clasificación binaria para cada par de clases. Por lo tanto, si se tienen c clases, el número total de modelos por entrenar es de $\frac{c(c-1)}{2}$. Al presentarse una observación nueva, se utiliza el sistema de votos a fin de obtener la clase predicha. El sistema de votos se basa en que dicha clase es la que obtenga más votos entre todos los clasificadores binarios (Franc y Hlavac, 2014).

Por consiguiente, la máquina de soporte vectorial para el problema multiclase debe de optimizar la siguiente función para encontrar el hiperplano con margen máximo (Xu et al, 2017):

$$min_{w_m b_m \xi_m} \frac{1}{2}(w_m^2)^T + C\sum_{i=1}^{n} \xi_{mi}$$

Donde w es el vector de pesos que representa el hiperplano de separación entre las dos clases. El peso asociado a cada variable predictora proporciona información sobre la relevancia para la discriminación entre las clases. C es un hiperparámetro de regularización, el cual puede ayudar a reducir el sobreajuste y, por ende, el error de clasificación. ξ_i es una variable que funciona para minimizar el error por medio de la distancia entre la observación y el hiperplano. ξ_i es igual a 0 para los casos que fueron separados correctamente por un margen suficiente. ξ_i se encuentra entre 0 y 1 en los casos que fueron separados correctamente, pero por un margen menor que el deseado por el modelo; mientras tanto, ξ_i es mayor a 1 para los casos que fueron clasificados incorrectamente. Finalmente, m representa cada par de modelos binarios.

La función anterior se encuentra sujeta a:

$$w_m^T x_i + b_m \geq 1 - \xi_{mi}, cuando\ y_i = 1$$

$$w_m^T x_i + b_m \leq -1 + \xi_{mi}, cuando\ y_i = -1$$

$$\xi_{mi} \geq 0$$

Donde $x_i$representan el vector que contiene el conjunto de n observaciones en el conjunto de datos de entrenamiento. y_i es la variable binaria por predecir, que puede tomar valores de 1 y -1. Por último, b es el intercepto o sesgo del hiperplano.

4.2.4.3 Validación de modelos

En esta sección se describe el método empleado para validar los modelos, y las medidas de precisión utilizadas para determinar su rendimiento.

4.2.4.3.1 Validación cruzada

Para validar los modelos se utilizó validación cruzada con k grupos. Esta técnica utiliza una parte de los datos para entrenar el modelo y otra parte para probarlo. Se debe dividir el conjunto de datos en k grupos del mismo tamaño. Se utiliza el grupo k para prueba y el resto de los datos se usa para entrenar el modelo. La ventaja de esta técnica consiste en que, luego, cada grupo k es utilizado como prueba, mientras que los restantes grupos son utilizados para entrenar el modelo. Como su nombre lo indica, este proceso se realiza k veces. Este método, al utilizar datos nuevos o de prueba, evita obtener resultados que podrían ser engañosos debido al sobreajuste de los modelos (Hastie et al., 2001).

$$CV_k = \frac{1}{k}\sum_{k=1}^{k} error_k$$

Para este trabajo en específico, se utilizó k = 10. El procedimiento de validación cruzada puede realizarse solo una vez para medir el error obtenido en la clasificación, o puede

efectuarse repetidas veces para obtener una mayor certeza y confianza del valor del error. El procedimiento de validación cruzada se realizó un total de cinco veces.

4.2.4.3.2 Medidas de ajuste

La medida que se utilizó para seleccionar el mejor modelo predictivo para cada una de las preguntas fue la precisión global. Con el modelo elegido, se calculó la precisión para cada una de las categorías j. Con el fin de evitar que las métricas mencionadas se vieran afectadas por el sobreajuste de los modelos, estas se estimaron a partir de la validación cruzada y sus repeticiones.

La precisión del modelo se encuentra dada por:

$$Precisión = \frac{Clasificados\ correctamente}{Clasificados\ correctamente + Clasificados\ incorrectamente}$$

La medida utilizada para seleccionar el mejor modelo para cada pregunta es la precisión estimada, a partir de la validación cruzada con diez grupos (k) y de las cinco repeticiones (i). Esta está es dada por:

$$Precisión = \frac{\sum_{i}^{5} \frac{\sum_{k}^{10} Precisión_{ik}}{10}}{5}$$

Por otra parte, se estimó la matriz de confusión a partir de las predicciones obtenidas en la validación cruzada y las repeticiones. En cada validación cruzada se obtuvo una predicción para cada respuesta. Como el proceso se repite cinco veces, la clase predicha se calculó con la moda de las clases obtenidas en cada repetición. Además, con el fin de comparar las clases predichas con las que fueron codificadas manualmente, se estimó la distribución porcentual y las precisiones para las clases predichas, por medio de los resultados obtenidos en la matriz de confusión.

La precisión para cada clase se calculó a partir de las categorías predichas. Estas se obtuvieron en la matriz de confusión mediante el cociente de los casos clasificados

correctamente en la clase j entre el total de los casos que fueron clasificados en la clase j por el modelo:

$$Precisión(j) = \frac{Clasificados\ correctamente(j)}{Predichos(j)}$$

Asimismo, por medio de un gráfico de dispersión, se estudió si existe relación entre la precisión de los modelos seccionados, el tamaño de muestra y el número de categorías de las preguntas.

4.3 Etapas del análisis de datos

Primeramente, se determina la cantidad de casos y el porcentaje de valores que no aplican para las 12 preguntas de análisis. Luego, se procede con la limpieza y el preprocesamiento de los datos textuales.

Una vez realizada la limpieza de los datos, se realizan los análisis exploratorios de las preguntas abiertas. También se realizan análisis de frecuencias y de nubes de palabras, utilizando unigramas y bigramas. En la misma línea del análisis exploratorio, se crea una red de palabras de bigramas, y se estiman medidas de centralidad. A continuación, se lleva a cabo el análisis de agrupamiento de palabras a partir del dendrograma con estructura rectangular y filogenética, para las palabras que se repiten cuatro veces o más. Para concluir con el análisis exploratorio, se redacta un breve resumen de los resultados obtenidos de las restantes once preguntas. Esto último se hace debido a que, en este documento, los análisis anteriores son presentados únicamente para una de las preguntas.

En el análisis de sentimientos, se crea un diccionario para el análisis de las preguntas. El diccionario contiene palabras consideradas como negativas y se indica su respectiva puntuación, de acuerdo con la magnitud de negatividad. Después, se procede a asignar la puntuación a cada una de las respuestas. Con esto se obtiene la distribución del puntaje de sentimiento para cada una de las preguntas.

Respecto al análisis de predicción de respuestas, primero, se convierten los datos de texto en una matriz documento-palabra. Luego, se eliminan las palabras que se repiten una única vez en la pregunta y las que no cumplen con el criterio establecido de la Chi-cuadrado. Una vez que se posee la estructura adecuada de análisis, se procede a entrenar los modelos para cada pregunta. La elección del modelo que predice las respuestas se determina mediante una validación cruzada con diez grupos y cinco repeticiones, y se estima la precisión global.

Seleccionado el modelo óptimo, se analiza si existe relación entre el tamaño de muestra, el número de categorías y la precisión para cada una de las preguntas. Se estima la matriz de confusión que se obtiene de la validación cruzada y sus repeticiones. Finalmente, se estiman las precisiones para cada una de las categorías y se compara la distribución porcentual de las clases predichas con la distribución de las clases que fueron codificadas manualmente.

A continuación, se presentan los resultados obtenidos al analizar las preguntas abiertas con los métodos propuestos.

CAPÍTULO V: RESULTADOS

En este capítulo se presentan los resultados obtenidos al analizar las respuestas de las preguntas abiertas de la ENTP - 2019. En una primera parte, se muestra un análisis exploratorio realizado por medio de la frecuencia de las palabras, del análisis de redes y del análisis de clusters. Seguidamente, se procede a mostrar los resultados obtenidos a partir del análisis de sentimientos. Por último, se presentan los modelos de predicción elegidos para cada una de las preguntas, la validación y una interpretación sustantiva de los resultados a partir de estos modelos.

El análisis de las doce preguntas genera una gran cantidad de resultados; por lo tanto, en esta sección se muestran, para los análisis exploratorios y predictivos, únicamente los resultados para la pregunta que hace referencia a los impedimentos que han surgido a la hora de participar en asuntos del sector público, la cual se incluye en el cuestionario de participación ciudadana.

5.1 Análisis exploratorio

En el Cuadro 1 se indica el tamaño de muestra inicial, la cantidad de muestra efectiva y el porcentaje que no sabe o no responde (NS/NR) para cada cuestionario y pregunta. En cada cuestionario, las personas que no respondieron la pregunta sobre barreras o impedimentos, no se les indagó sobre soluciones o qué harían desde su propio ámbito. Se aprecia que el porcentaje de observaciones denominadas como no sabe o no responde, varía entre 5% y 18%.

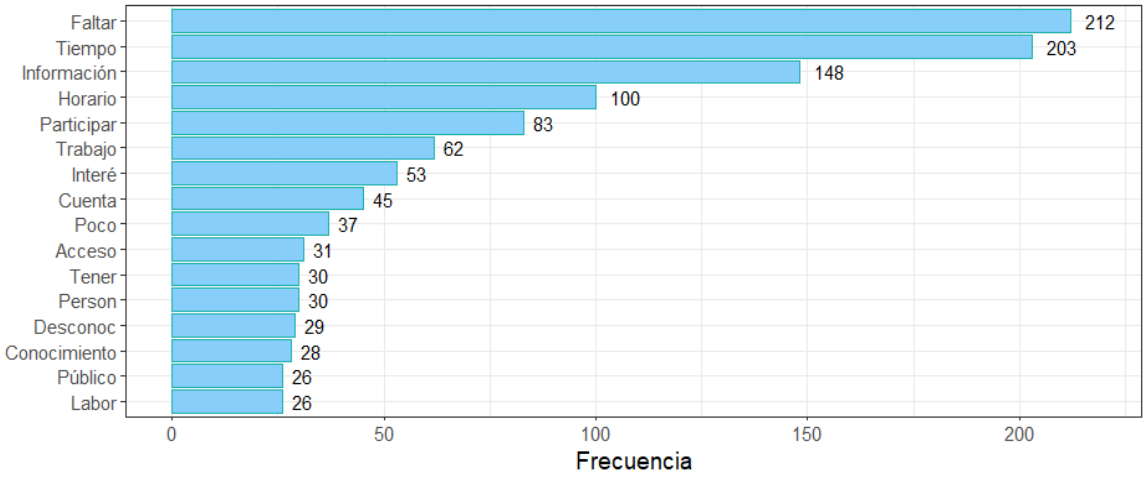
Cuadro 1. Tamaño de muestra, muestra efectiva y porcentaje de no sabe/no responde para cada pregunta

Pregunta	Muestra inicial[1]	Muestra efectiva[2]	Porcentaje de NS/NR
Acceso a la información			
Impedimentos	1105	995	9.95
Soluciones	825	768	5.17
¿Qué haría la persona?	825	622	18.38
Rendición de cuentas			
Impedimentos	1095	960	12.33
Soluciones	764	664	9.16
¿Qué haría la persona?	763	599	14.99
Participación ciudadana			
Impedimentos	1090	919	15.69
Soluciones	871	713	14.50
¿Qué haría la persona?	882	688	17.80
Funcionarios públicos			
Impedimentos	607	557	8.24
¿Qué haría si fuera jerarca?	529	458	11.70
Qué haría la persona?	520	437	13.71

Nota: [1]la muestra inicial representa la cantidad de personas a las que se les realizó la respectiva pregunta; eliminando los no aplica.
Nota: [2]la muestra efectiva resulta de eliminar los NS/NR.
Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

Lo primero que se hizo en el análisis fue obtener cuáles son las quince palabras que más se repiten. Para ello, se estimó la frecuencia de cada una de las palabras para la pregunta que se refiere a los impedimentos a la hora de participar en asuntos del sector público. En el Gráfico 1 se observa que las palabras con una mayor frecuencia son: ‘faltar’ (212), ‘tiempo’ (203), ‘información’ (148) y ‘horario’ (100). Además, entre las palabras más frecuentes aparecen: ‘participar’, ‘trabajo’, ‘poco’, ‘acceso’, ‘conocimiento’, ‘público’ y ‘labor’.

Gráfico 1. Frecuencia de palabras que más ocurren, en referencia a la pregunta sobre los impedimentos para participar en asuntos del sector público



Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

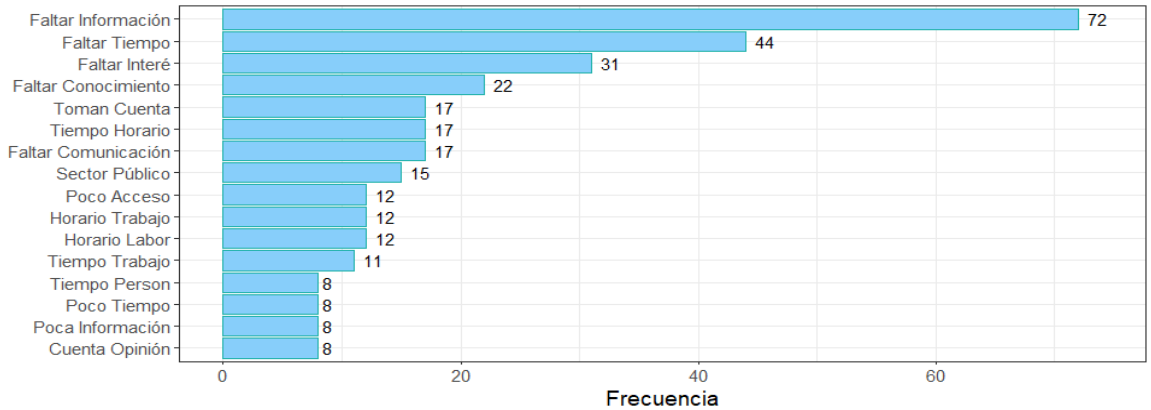
Otra manera de visualizar la frecuencia de las palabras en el texto es mediante una nube de palabras. A diferencia del Gráfico 1, en la Figura 3 se pueden observar más palabras que las quince más frecuentes incluidas en aquel. En la Figura 3, se aprecian palabras que aparecen en menor medida en las respuestas que se refieren a los impedimentos para participar en asuntos del sector público. Algunas de estas palabras son: ‘burocracia’, ‘corrupción’, ‘desconocimiento’, ‘trámites’, ‘lejanía’, ‘edad’ y ‘argolla’.

Figura 3. Nube de palabras que hace referencia a los impedimentos para participar en asuntos del sector público

Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

La frecuencia de los bigramas que más aparecen en las respuestas se muestra en el Gráfico 2. Los que más se repiten son: ‘falta de información’ (72), ‘falta de tiempo’ (44), ‘falta de interés’ (31) y ‘falta de conocimiento’ (22). Asimismo, entre los bigramas más frecuentes se encuentran: ‘tomar en cuenta’, ‘tiempo horario’, ‘falta de comunicación’, ‘poco acceso’, ‘horario de trabajo’, ‘poco tiempo’, ‘poca información’ y ‘cuenta opinión’.

Gráfico 2. Frecuencia de bigramas que más ocurren en referencia a los impedimentos para participar en asuntos del sector público



Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

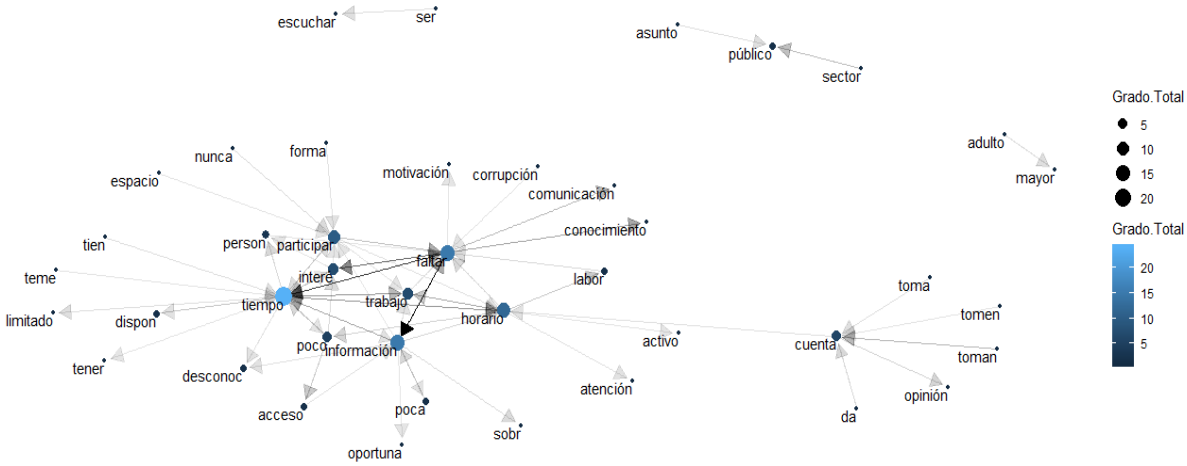
En la Figura 4 se aprecia la nube de palabras formada por bigramas. En esta, aparecen impedimentos como: ‘tiempo limitado’, ‘ser escuchado’, ‘acceso a la información’, ‘ser adulto mayor’, ‘falta de motivación’, ‘información oportuna’ y ‘nunca participó’.

Figura 4. Nube de bigramas que hace referencia a los impedimentos para participar en asuntos del sector público

Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

Por otra parte, una alternativa para visualizar los bigramas es mediante un análisis de redes. En la red se puede apreciar la centralidad que presenta cada nodo o palabra por medio del indicador de grado total. El gradiente de la flecha en esta red representa la frecuencia del bigrama en las respuestas: cuanto más oscura sea la flecha, más se repite el bigrama. Mediante este procedimiento, se logra evidenciar que la palabra ‘tiempo’ es la que presenta una mayor centralidad en la red; a esta le siguen las palabras: ‘faltar’, ‘información’, ‘participar’ y ‘horario’ (Figura 5).

Figura 5. Red de bigramas[1] que hacen referencia a los impedimentos para participar en asuntos del sector público



Nota: [1]en la red se muestran los bigramas que se repiten tres veces o más.
Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

El Cuadro 2 muestra los indicadores de nodo de grado total, grado interno y grado externo. Este cuadro permite reafirmar que las palabras que reciben y emiten más conexiones son: ‘tiempo’ (24), ‘faltar’ (15) e ‘información’ (15). La palabra ‘tiempo’ puede aparecer antes de una palabra (13) o seguida de otra (11), mientras que la palabra ‘información’ suele aparecer mayormente antes de otra palabra (10).

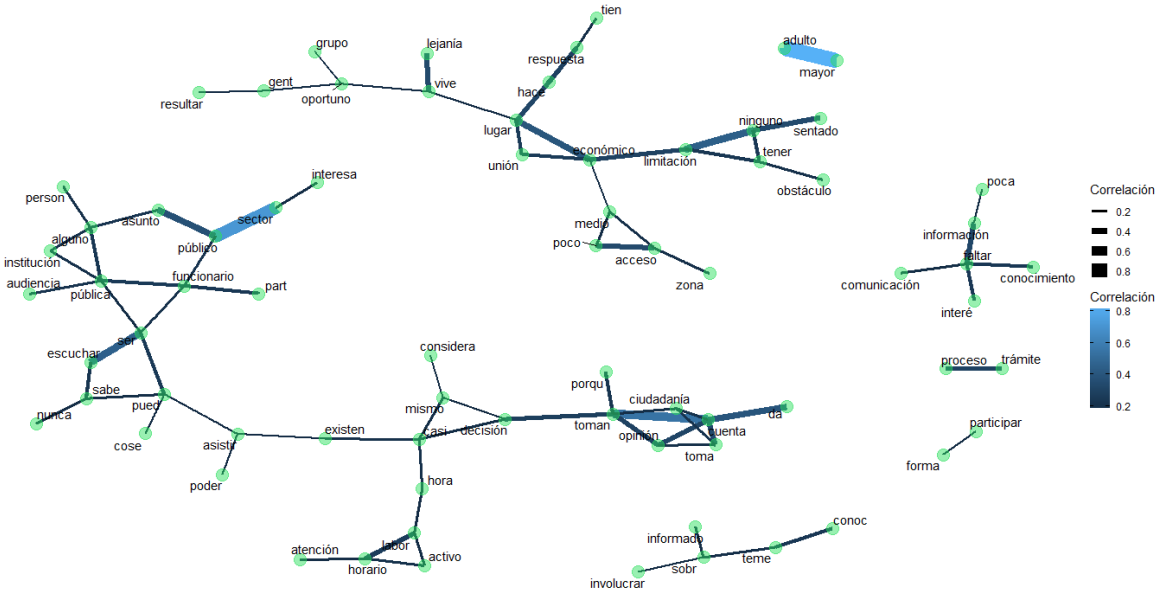
Cuadro 2. Indicadores de grado del nodo para la red de bigramas

Palabra	Grado total	Grado interno	Grado externo
Tiempo	24	11	13
Faltar	15	9	6
Información	15	5	10
Horario	12	5	7
Participar	10	6	4
Interé	7	3	4
Trabajo	7	4	3
Cuenta	6	4	2
Poco	5	2	3
Person	3	2	1
Poca	2	1	1
Labor	2	1	1
Dispon	2	1	1
Acceso	2	1	1
Público	2	2	0

Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

De forma complementaria al análisis de redes en palabras, se analizaron las correlaciones entre estas por medio del coeficiente phi. En la red, el gradiente de color y el grosor de la línea de conexión representan la magnitud de la correlación. En la Figura 6, se aprecia que dos palabras que se encuentran altamente correlacionadas son ‘adulto’ y ‘mayor’, al igual que ‘sector’ y ‘público’. Además, se logra observar la conexión por medio de la magnitud de las correlaciones de otras palabras como: ‘toman’ y ‘cuenta’, ‘ninguna’ y ‘limitación’, y ‘ser’ y ‘escuchado’.

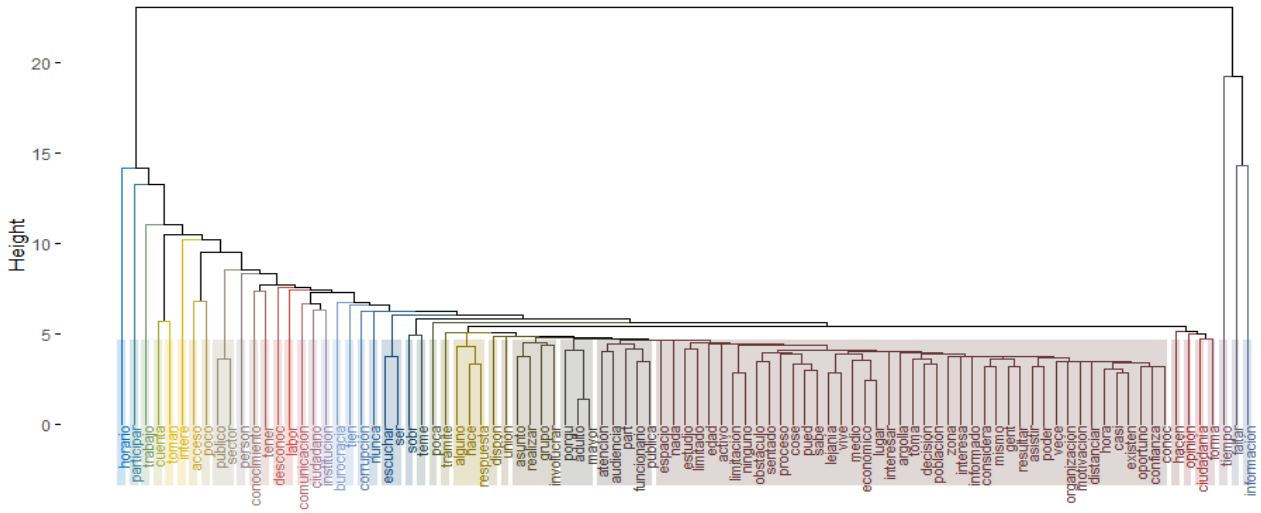
Figura 6. Red de correlaciones[1] entre palabras que hacen referencia a los impedimentos para participar en asuntos del sector público



Nota: [1]la red muestra las correlaciones mayores o iguales a 0,16.
Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

Seguidamente, con base en la exploración descriptiva de los datos, se presentan los resultados a partir del análisis de clusters jerárquicos. En este análisis se muestra la agrupación de las palabras en conglomerados, según sus distancias o similitudes. En la Figura 7, se puede distinguir el dendrograma rectangular que hace referencia a las respuestas sobre impedimentos a la hora de participar en asuntos del sector público. En este, se evidencia que las palabras que se unieron de primero, por presentar menor distancia, son ‘adulto’ y ‘mayor’. Luego, al agregar la palabra ‘porque’ en el grupo, se puede interpretar que el impedimento al que se hace referencia es ‘porque se es adulto mayor’. Adicionalmente, se forman otros conglomerados: ‘vive lejos’, ‘estudio limitado’, ‘poco acceso’, ‘funcionarios públicos’, entre otros.

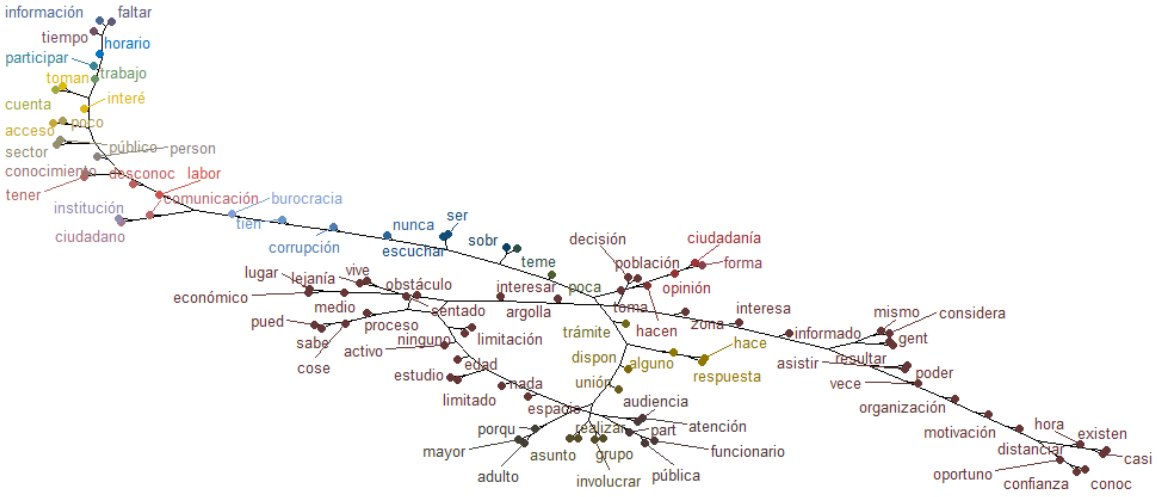
Figura 7. Dendrograma rectangular de palabras[1] que hacen referencia a los impedimentos para participar en asuntos del sector público



Nota: [1]el dendrograma muestra las palabras que se repiten 4 veces o más.
Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

Otra forma de representar el análisis de conglomerados es mediante un árbol filogenético, como lo muestra la Figura 8. En esta, se observa en cada rama cómo se forman los conglomerados de palabras según sus distancias.

Figura 8. Dendrograma filogenético de palabras[1] que hacen referencia a los impedimentos para participar en asuntos del sector público



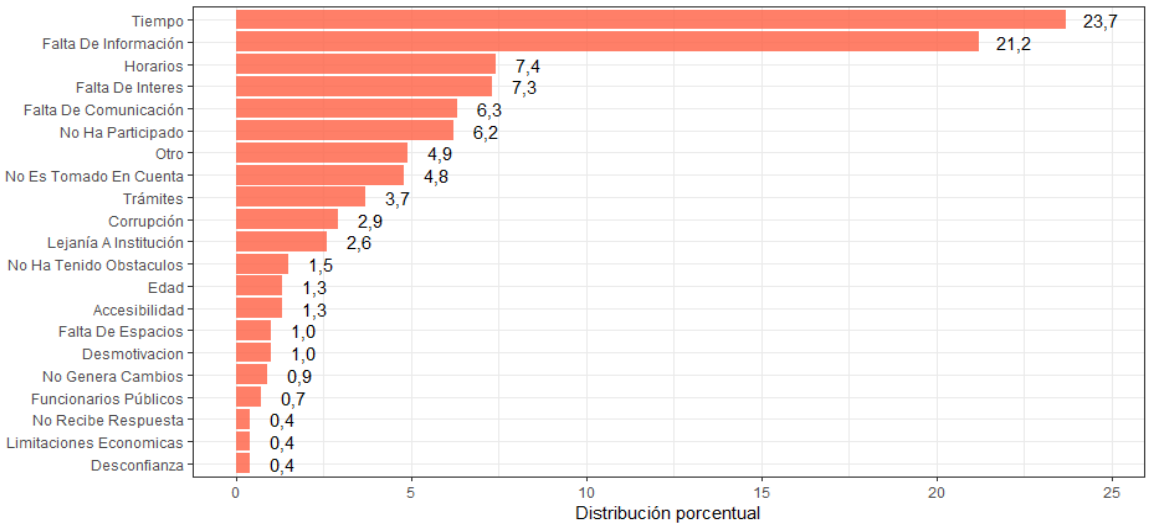
Nota: [1]el dendrograma muestra las palabras que se repiten 4 veces o más.
Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

El siguiente apartado tiene como finalidad comparar los resultados obtenidos mediante el análisis exploratorio de las respuestas. Para ello, se emplean técnicas provenientes de la minería de texto con los resultados y se utilizan las categorías que fueron establecidas manualmente.

5.1.1 Comparación del análisis exploratorio con las clases codificadas manualmente

A continuación, se procede a mostrar la distribución porcentual de las categorías preestablecidas manualmente para la pregunta de la ENTP – 2019 que hace referencia a los impedimentos para participar en asuntos del sector público.

Gráfico 3. Distribución porcentual de las clases codificadas manualmente para la pregunta que hace referencia a los impedimentos para participar en asuntos del sector público



Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

El Gráfico 3 muestra que, según la codificación manual, las personas indicaron que tienen como barrera el tiempo (24%), la falta de información (21%), los horarios (8%), la falta de interés (7%) y falta de comunicación (6%). Al observar los análisis exploratorios mostrados anteriormente utilizando técnicas de minería de texto, estas mismas categorías son representadas en las frecuencias de unigramas y bigramas, como las más mencionadas por las personas entrevistadas. Categorías como ‘corrupción’, ‘trámites’, ‘edad’, ‘lejanía’, ‘la accesibilidad’ y ‘no haber participado’ también se observan en el análisis al utilizar técnicas provenientes de la minería de datos.

5.1.2 Resumen de las demás preguntas

Este apartado pretende mostrar una breve descripción de los resultados obtenidos, mediante el análisis exploratorio, al utilizar técnicas de minería de texto para las restantes once preguntas de la ENTP - 2019[12].

Se encontró que, al consultar a las personas cómo se podría solucionar esa barrera, mencionaron, entre otras posibilidades, mejorar los sitios web y las redes sociales, actualizar la información, aumentar la cantidad de información que ofrecen y mejorar el acceso. Respecto a la pregunta de qué podría hacer la persona desde su propio ámbito, las personas respondieron lo siguiente: participar, buscar información, solicitar información, obtener provecho de los sitios web o no hacer nada.

Cuando se les preguntó a los funcionarios públicos qué harían, si fueran jerarcas, para eliminar o solucionar las barreras, mencionaron: dar más información, mejorar la rendición de cuentas, y brindar información por medios de comunicación y páginas web. Por otra parte, al preguntarles qué podrían hacer desde su propio ámbito, señalaron: brindar la información, nada, ser transparentes, actualizar la información, ser honestos y ofrecer un mejor servicio.

A continuación, se procede a mostrar los resultados obtenidos por medio de la aplicación del análisis de sentimientos.

5.2 Análisis de sentimientos

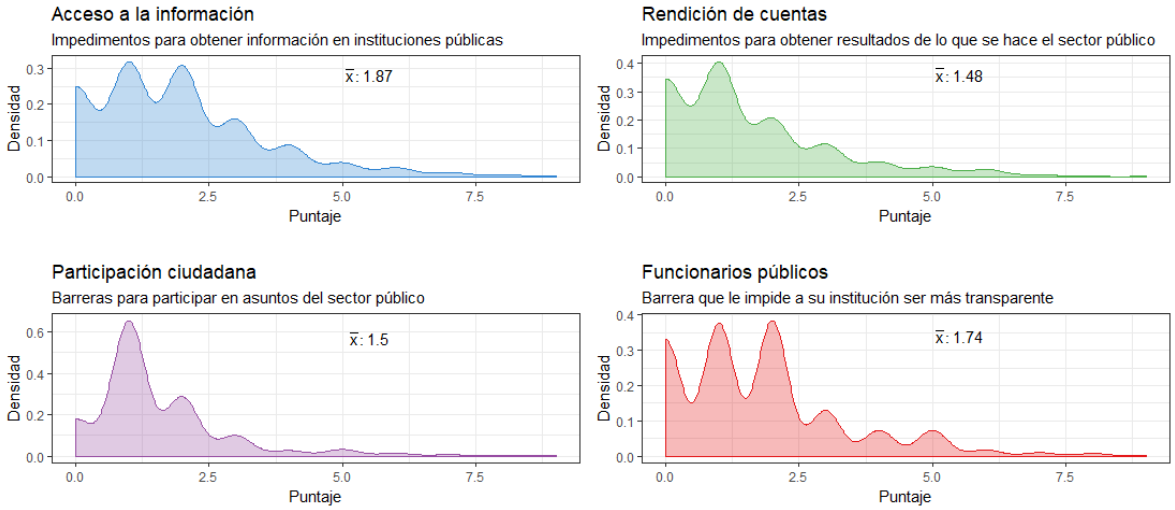
El Gráfico 4 muestra los resultados obtenidos a partir del análisis de sentimientos de las cuatro preguntas analizadas referentes a los impedimentos y barreras en la obtención de información, conseguir resultados y a la participación en asuntos del sector público. Los datos evidencian que las cuatro distribuciones del puntaje de sentimiento presentan una asimetría negativa, lo que indica que son pocas las respuestas que obtuvieron un puntaje de sentimiento negativo alto. Al comparar las cuatro distribuciones entre sí, las respuestas del cuestionario

[12] Las preguntas se pueden observar en la Tabla 1, que se encuentra en la sección de materiales.

de participación ciudadana se centran en puntajes de 1. Por su parte, la distribución de la pregunta de acceso a la información muestra una media mayor en comparación con las demás: su promedio es de 1,87.

A manera de ejemplo, una respuesta que obtuvo un puntaje de 9 en la pregunta relativa a impedimentos para participar en asuntos del sector público es: “No fui atendido en la CCSS durante la huelga y no recibí respuesta, falta de servicio al cliente y luego falta de disposición para atender. Demasiada info pero no es clara ni accesible”. Con esto, se evidencia que la respuesta contiene varias palabras que son consideradas como negativas en el lexicón creado.

Gráfico 4. Distribución de puntajes obtenidos a partir del análisis de sentimientos de respuestas que hacen referencia a los impedimentos y barreras a componentes de la transparencia



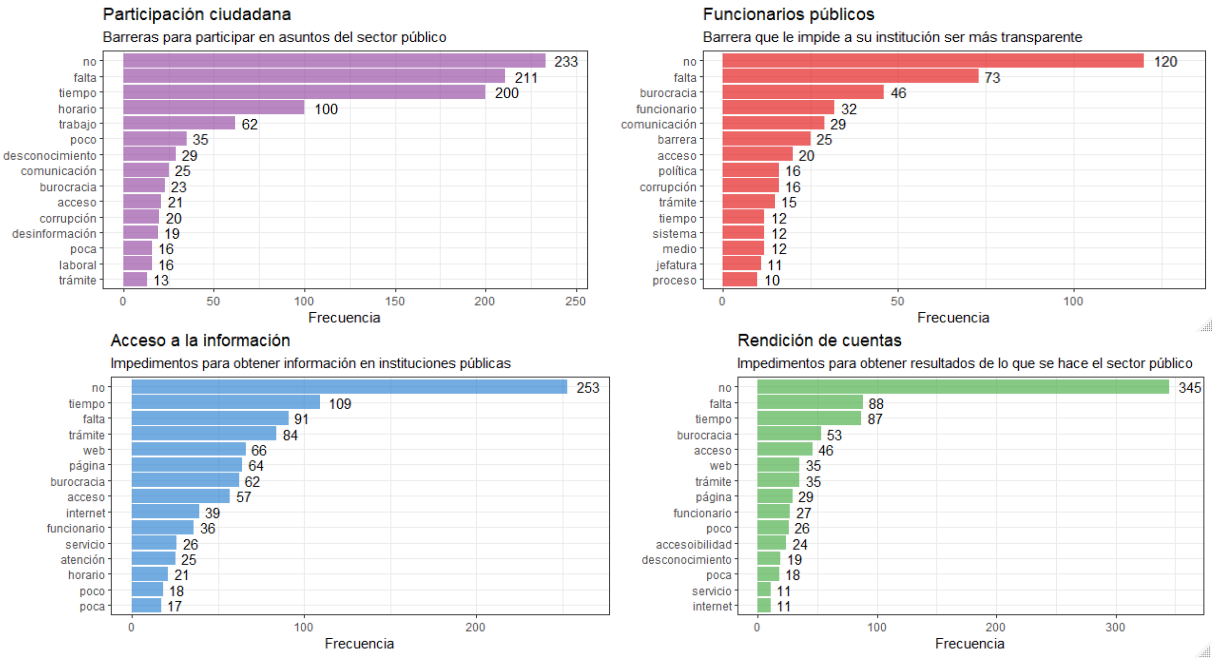
Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

El Gráfico 5 muestra las palabras más frecuentes, incluidas en el lexicón creado para las preguntas analizadas, que aparecen en las respuestas. La palabra ‘no’ es la que más se repite en las respuestas de las cuatro preguntas; en el cuestionario de rendición de cuentas fue en el que se mencionó en más ocasiones (345). Otras palabras que son de las más

frecuentes y que son comunes en las respuestas correspondientes a las cuatro preguntas son: ‘falta’, ‘tiempo’ y ‘burocracia’.

Las palabras de mayor frecuencia, y que poseen una mayor ponderación en el lexicón creado en las cuatro preguntas, son: ‘corrupción’ (5 de puntaje), ‘trámite’ (2), ‘burocracia’ (2), ‘página’ (2), ‘web’ (2), ‘funcionario’ (2), ‘política’ (2) y ‘jefatura’ (2). Palabras con los pesos mayores de negatividad como ‘negar’, ‘censurar’, ‘confidencial’, ‘esconden’, ‘desactualizada’, ‘falsa’, ‘ignoran’, ‘negligencia’ y ‘pésimo’ no aparecen entre las más frecuentes.

Gráfico 5. Palabras más frecuentes obtenidas a partir del análisis de sentimientos de respuestas que hacen referencia a los impedimentos y barreras a componentes de la transparencia



Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

Seguidamente, se procede a mostrar los resultados obtenidos al realizar la clasificación de las respuestas mediante la utilización de técnicas de aprendizaje automático supervisado.

5.3 Modelos predictivos de las respuestas

Esta sección expone los resultados generales para las doce preguntas analizadas, al aplicar los métodos predictivos. Además, se muestran los resultados relativos a la validación e interpretación de las predicciones para la pregunta que hace referencia a los impedimentos para participar en asuntos del sector público.

5.3.1 Resultados generales

El Cuadro 3 muestra los algoritmos seleccionados para cada pregunta, esto según su precisión al realizar las validaciones cruzadas. El clasificador ingenuo de Bayes fue el que se seleccionó en más ocasiones, seguido de los bosques aleatorios. Las máquinas de soporte vectorial fueron elegidas en dos ocasiones; al considerar la pregunta sobre cómo solucionaría la barrera, incluida en el cuestionario de acceso a la información, y la pregunta realizada a los funcionarios públicos sobre qué harían desde su propio ámbito para solucionar la barrera.

En cuanto a la precisión de los modelos, la pregunta en la que se consiguió una mayor precisión, a partir del uso de la validación cruzada, fue la que hace referencia a los impedimentos al participar en asuntos del sector público, pregunta incluida en el cuestionario de participación ciudadana. Dicha pregunta obtuvo un 76,3%, y fue seguida por la pregunta sobre barreras para obtener resultados de lo que hace el sector público, con un 74,4%. Esta última pregunta está incluida en el cuestionario de rendición de cuentas. La pregunta referente a qué haría si fuera jerarca para solucionar las barreras, correspondiente al cuestionario de funcionarios públicos, fue en la que se obtuvo una precisión menor a la hora de predecir las categorías (48%).

Además, los resultados obtenidos muestran la cantidad de palabras únicas[13] iniciales y el total de palabras finales que se utilizaron para entrenar los modelos. Se evidencia que la dimensión de los predictores para entrenar los modelos disminuyó considerablemente al

[13] El que sean palabras únicas indica que el conteo hace referencia a palabras distintas. A manera de ejemplo, si la palabra ‘trámite’ aparece 50 veces en las respuestas a una pregunta en específico, en el conteo aparece solo una vez.

eliminar palabras que no eran relevantes para predecir la categoría de las respuestas (Cuadro 3).

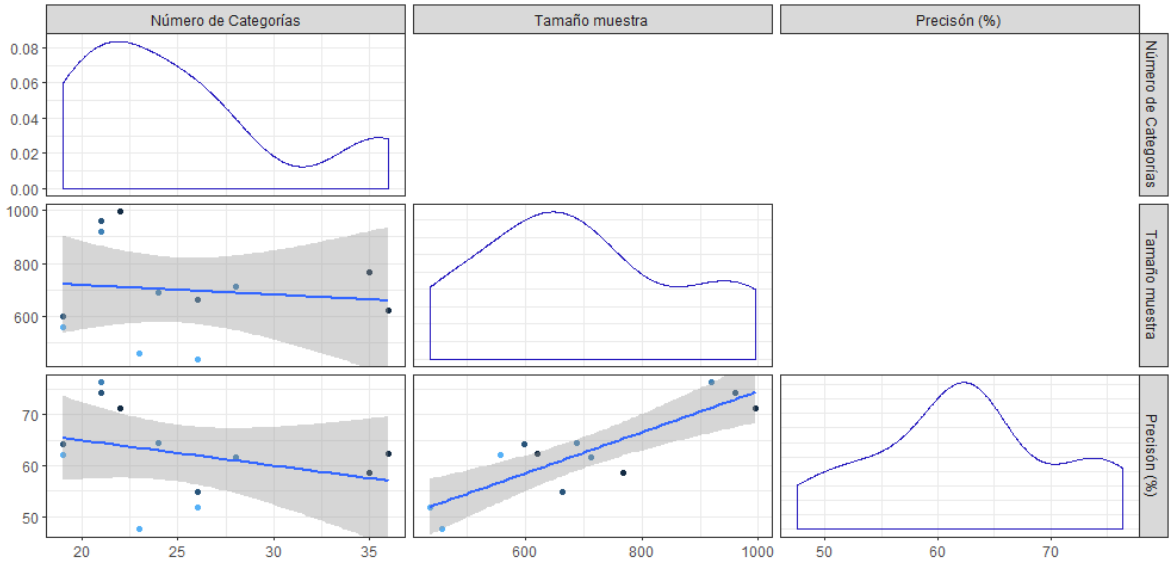
Cuadro 3. Modelos de predicción seleccionados para cada pregunta, número de predictores iniciales y finales y su respectiva precisión

Pregunta	Modelo elegido	Número de categorías	Total de palabras iniciales	Total de palabras finales[1]	Precisión[2]
Acceso a la información					
Impedimentos	Bayes	22	590	108	71,37
Soluciones	SVM - Lineal	35	708	100	58,56
¿Qué haría la persona?	Bosques aleatorios	36	602	160	62,25
Rendición de cuentas					
Impedimentos	Bayes	21	578	139	74,41
Soluciones	Bosques aleatorios	26	725	113	54,78
¿Qué haría la persona?	Bosques aleatorios	19	590	137	64,17
Participación ciudadana					
Impedimentos	Bosques aleatorios	21	599	241	76,34
Soluciones	Bayes	28	806	215	61,59
¿Qué haría la persona?	Bayes	24	691	196	64,58
Funcionarios públicos					
Impedimentos	Bayes	19	582	227	62,20
¿Qué haría si fuera jerarca?	Bayes	23	767	218	47,50
¿Qué haría la persona?	SVM - Sigmoideo	26	622	157	51,79

Nota: [1]las palabras finales son las que se obtienen al eliminar las que se repiten una sola vez, y con la utilización de la prueba Chi-cuadrado.
Nota: [2]la precisión se obtuvo a partir del promedio de las precisiones en la validación cruzada, considerando diez grupos y cinco repeticiones.
Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

Por otra parte, se estudió la relación entre la precisión obtenida, el tamaño de muestra y el número de categorías. El Gráfico 6 destaca que, entre las preguntas, existe una relación lineal positiva entre el tamaño de muestra utilizado para entrenar los modelos y la precisión obtenida. Lo anterior indica que, al aumentar el tamaño de muestra, la precisión aumenta. Se observa, además, una relación lineal negativa entre el número de categorías de las preguntas y la precisión obtenida.

Gráfico 6. Relación entre la precisión obtenida, el tamaño de muestra y el número de categorías de las doce preguntas

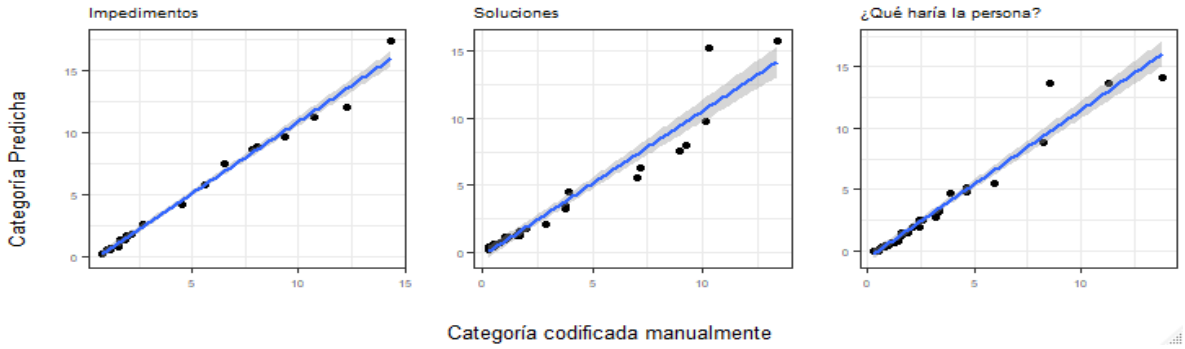


Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

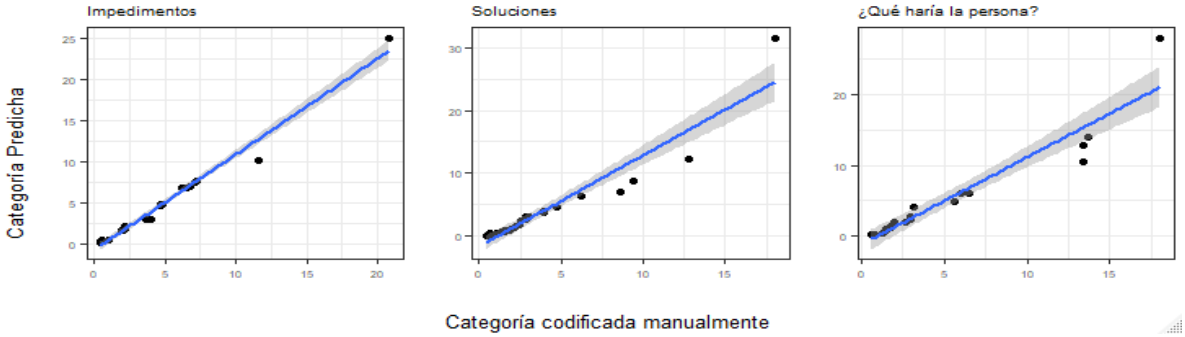
Se realizó un gráfico de dispersión con el propósito de apreciar la relación existente entre la distribución porcentual de las categorías obtenidas mediante la codificación manual y la distribución de las categorías predichas para cada una de las preguntas de los cuatro cuestionarios. El gráfico resultante muestra una relación lineal positiva entre ambas distribuciones, inclusive en la pregunta del cuestionario de funcionarios públicos que hace referencia a qué harían estos si fueran jerarcas, en la que se obtuvo una precisión del 48% (Gráfico 7). El que se hayan encontrado correlaciones relativamente fuertes entre ambas distribuciones no indica que se obtienen conclusiones sustantivas similares entre las categorías codificadas manualmente y las predichas: para determinar esto se deben analizar y comparar los valores de la distribución porcentual, como lo muestra más adelante el Cuadro 4.

Gráfico 7. Relación entre distribución porcentual de las categorías codificadas manualmente y distribución de las categorías predichas

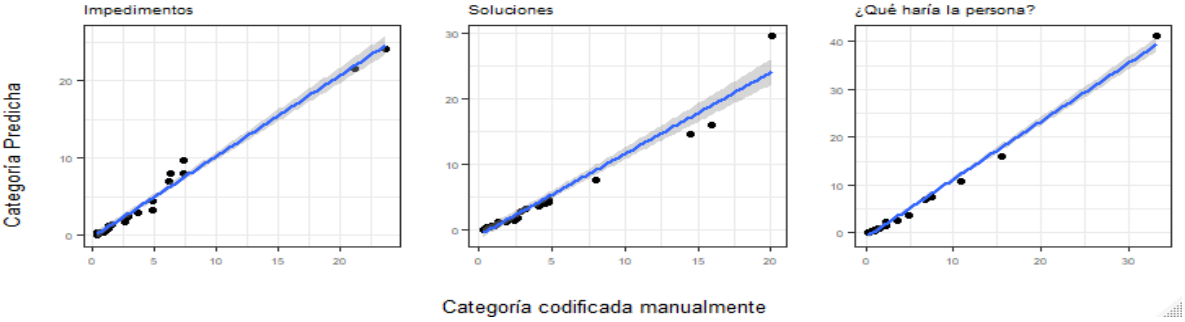
Acceso a la información



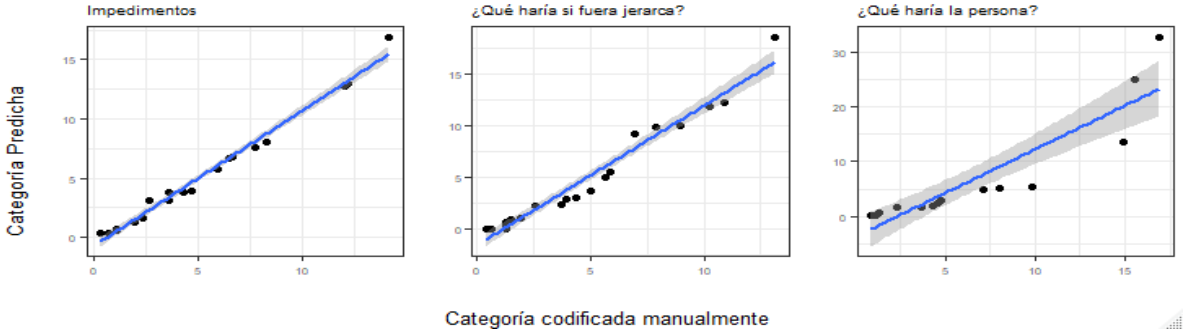
Rendición de cuentas



Participación ciudadana



Funcionarios públicos



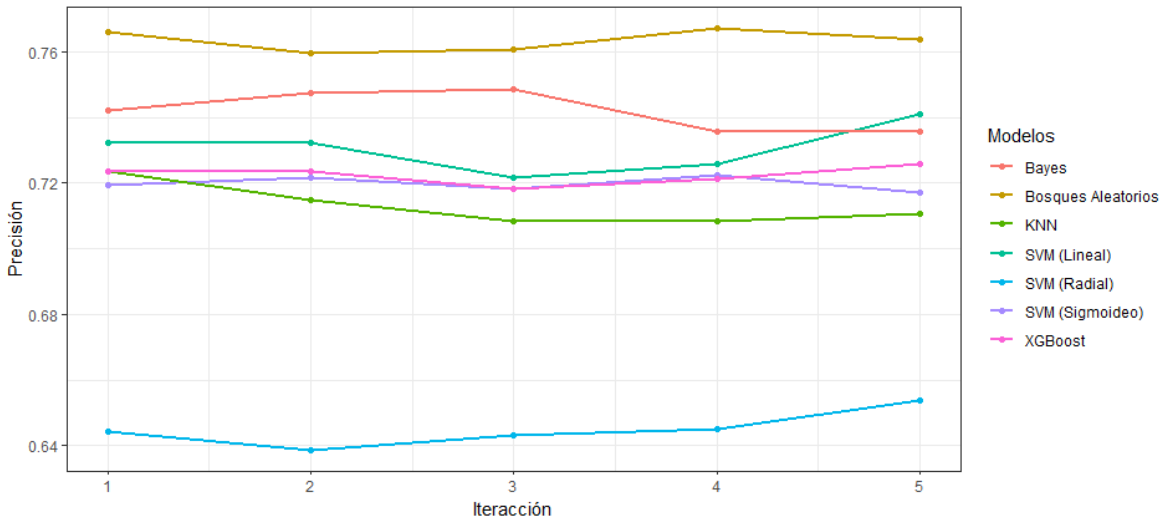
Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

A continuación, se presenta cómo se realizó la selección, validación e interpretación de los resultados del modelo seleccionado; esto para la pregunta que se refiere a los impedimentos para participar en asuntos del sector público, incluida en el cuestionario de participación ciudadana. Se debe señalar que el mismo procedimiento de selección y validación se realizó para las restantes preguntas; sin embargo, a manera de ejemplo, se muestra para una única pregunta.

5.3.2 Resultados para la pregunta referente a impedimentos para participar en asuntos públicos

Para elegir el mejor modelo para esta pregunta, se efectuó una validación cruzada que consideró diez grupos y cinco repeticiones. Los resultados que comparan los modelos entrenados se muestran en el Gráfico 8. En este, se aprecia que el modelo que tiene una mayor precisión en las cinco validaciones cruzadas son los bosques aleatorios; por ende, este algoritmo es el que se escogió para estimar las predicciones de las respuestas.

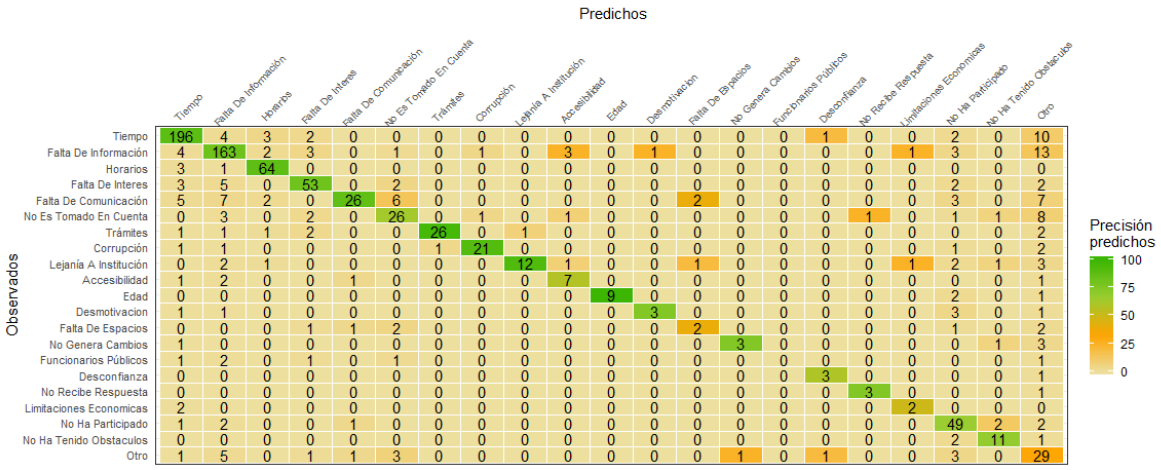
Gráfico 8. Repeticiones de validación cruzada para los modelos propuestos para la pregunta que hace referencia a los impedimentos para participar en asuntos del sector público



Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

En el Gráfico 9 se aprecia la matriz de confusión obtenida con el modelo de bosques aleatorios. En dicha matriz, se observa cómo clasificó el algoritmo las respuestas y la gradiente de color que representa la precisión de las categorías predichas. El modelo clasificó 90 respuestas en la categoría “otros”, de las cuales únicamente 29 pertenecen a dicha categoría. Las restantes 61 respuestas pertenecen a otras categorías.

Gráfico 9. Matriz de confusión[1] obtenida a partir de la predicción de los bosques aleatorios para la pregunta que hace referencia a los impedimentos para participar en asuntos del sector público



Nota: [1]los datos de la matriz de confusión fueron estimados a partir de los resultados de la validación cruzada y las repeticiones.
Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

Por otro lado, el Cuadro 4 muestra la comparación entre la distribución porcentual de las categorías codificadas manualmente y las predichas por los bosques aleatorios. Para las categorías preestablecidas, las dos clases con mayor frecuencia son las que indican que su mayor impedimento son el ‘tiempo’ (23,7%) y la ‘falta de información’ (21,2%). El resultado obtenido fue el mismo para las clases predichas: las dos clases con mayor frecuencia son el tiempo (24,0%) y la falta de información (21,6%). Las categorías de ‘horarios’, ‘falta de interés’ y ‘falta de comunicación’ representan 7,4%, 7,3% y 6.3%, respectivamente. Por otra parte, estas mismas categorías para las clases predichas representan 7,9%, 7,1% y 3,3%, respectivamente. Asimismo, se observa una similitud entre la distribución porcentual de las

categorías codificadas manualmente que presentan porcentajes menores y sus respectivas categorías predichas.

Las precisiones de las categorías predichas para las clases de ‘edad’, ‘trámites’ y ‘corrupción’ son mayores a 91%, mientras que para las categorías ‘tiempo’ y ‘falta de información’, las precisiones respectivas son 88,7% y 81,9%. Las categorías que muestran precisiones menores son las de ‘funcionarios públicos’, ‘otro’ y ‘falta de espacios’.

Cuadro 4. Comparación de clases codificadas manualmente con las clases predichas[1] por los bosques aleatorios para la pregunta que hace referencia a los impedimentos para obtener información de instituciones públicas

Codificación manual		Predicho		
Categoría	**Distribución porcentual (%)**	**Categoría**	**Distribución porcentual (%)**	**Precisión (%)**
Tiempo	23,72	Tiempo	24,05	88,69
Falta de información	21,22	Falta de información	21,65	81,91
Horarios	7,40	Otro	9,79	32,22
Falta de interés	7,29	No ha participado	8,05	66,22
Falta de comunicación	6,31	Horarios	7,94	87,67
No ha participado	6,20	Falta de interés	7,07	81,54
Otro	4,90	No es tomado en cuenta	4,46	63,41
No es tomado en cuenta	4,79	Falta de comunicación	3,26	86,67
Trámites	3,70	Trámites	2,94	96,30
Corrupción	2,94	Corrupción	2,50	91,30
Lejanía a institución	2,61	No ha tenido obstáculos	1,74	68,75
No ha tenido obstáculos	1,52	Lejanía a institución	1,41	92,31
Accesibilidad	1,31	Accesibilidad	1,31	58,33
Edad	1,31	Edad	0,98	100,00
Desmotivación	0,98	Falta de espacios	0,54	40,00
Falta de espacios	0,98	Desconfianza	0,54	60,00
No genera cambios	0,87	Desmotivación	0,44	75,00
Funcionarios públicos	0,65	No genera cambios	0,44	75,00
Desconfianza	0,44	Limitaciones económicas	0,44	50,00
Limitaciones económicas	0,44	No recibe respuesta	0,44	75,00
No recibe respuesta	0,44	Funcionarios públicos	0,00	0,00

Nota: [1]la distribución porcentual y la precisión para las clases predichas se obtuvieron por medio de los datos de la matriz de confusión, a partir de los resultados de la validación cruzada y sus repeticiones.
Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

En el *anexo I* se muestra la comparación entre la distribución de las clases que fueron codificadas manualmente y las predichas, para las restantes once preguntas.

CAPÍTULO VI: CONCLUSIONES Y DISCUSIÓN

El presente trabajo mostró la aplicación tanto del análisis como de la automatización de la codificación a preguntas abiertas de la ENPT - 2019. Se partió de que las aplicaciones de la minería de texto ofrecen una alternativa analítica para datos extraídos a partir de preguntas abiertas en una encuesta de opinión. La primera etapa en el análisis texto efectuado fue la limpieza y el preprocesamiento de la información. Durante el proceso de análisis, se mostró la utilización de técnicas de análisis descriptivas, como lo son las frecuencias de palabras (nubes), redes de texto, análisis de correlación, clusters y análisis de sentimientos; asimismo, se mostró la utilización de técnicas predictivas de aprendizaje automático supervisado.

El empleo de las técnicas provenientes de la minería de texto demostró que es posible realizar un análisis del texto de manera automatizada, a diferencia de lo que ocurre en la codificación manual de las preguntas abiertas, en la que el proceso es subjetivo y conlleva una gran cantidad de tiempo. Se demostró, también, que las técnicas utilizadas en este trabajo permiten visualizar e identificar rápidamente los principales temas o conceptos presentes en las respuestas y las relaciones entre las palabras de una forma explícita.

Entre los resultados más importantes producto del análisis descriptivo se destaca el siguiente: las principales barreras o impedimentos que las personas encuentran a la hora de obtener información y participar en asuntos del sector público son, entre otros, la gran cantidad de trámites, el tiempo, el desconocimiento y la complicación al usar las páginas web. Al preguntarles a las personas cómo se podrían eliminar esas barreras, mencionaron soluciones como las siguientes: simplificando trámites, hacer eficiente el proceso, y por medio de mejorar y actualizar los sitios web.

En lo que respecta a la predicción de respuestas, se mostró que la tarea de codificación automática puede plantearse como un problema de categorización multi-clase por medio del aprendizaje automático supervisado. Se presentó una alternativa para seleccionar el número de palabras elegidas para entrenar los modelos, lo que disminuye considerablemente la

dimensionalidad. Además, se mostró cómo seleccionar el modelo más adecuado para cada una de las preguntas mediante validación cruzada. Los algoritmos que fueron seleccionados con mayor ocurrencia para las doce preguntas fueron el clasificador ingenuo de Bayes y los bosques aleatorios. La precisión resultante para cada uno de los modelos seleccionados para cada pregunta varió entre 48% y 76%.

A partir de los resultados obtenidos de los modelos predictivos, se encontró que existe una relación lineal positiva entre el tamaño de muestra y la precisión: a mayor tamaño de muestra, mayor es la precisión. Estos resultados permiten concluir que, en algunas preguntas, al incluirse tantas categorías, se debe aumentar el tamaño de muestra para incrementar las precisiones de los modelos, o se deben crear menos categorías, pues a veces no es fácil aumentar el tamaño de muestra. Por otro lado, al observar las matrices de confusión, se observó el ruido que añade la categoría "otros" en los modelos, debido a que las respuestas en esa categoría no siguen ningún patrón.

Los resultados obtenidos a partir de técnicas exploratorias como la frecuencia de palabras, análisis de redes y clusters son similares a los resultados que fueron codificados manualmente. Por otra parte, las categorías predichas por los modelos elegidos para cada pregunta permiten establecer resultados similares en comparación a las categorías preestablecidas. Esto permite concluir que los resultados obtenidos con la codificación manual y por medio de la utilización de técnicas de la minería de texto, tanto exploratorias como predictivas, son semejantes.

Finalmente, se creó una aplicación web utilizando Shiny, la cual permitió visualizar los resultados obtenidos en este documento de una manera interactiva: se logró cambiar los parámetros en el análisis de frecuencias, redes y clusters. Además, con la aplicación construida se creó un sistema que permite hacer predicciones en tiempo real para la pregunta que hace referencia a los impedimentos para participar en asuntos del sector público; también se creó un sistema que devuelve el puntaje de sentimiento para una respuesta dada.

Para investigaciones futuras, al seleccionar las palabras con las que se entrenen los modelos, se recomienda crear una función que: en primer lugar, alterne los puntos de corte de la probabilidad asociada correspondiente a la Chi-cuadrado; en segundo lugar, estime los algoritmos por comparar; en tercer lugar, valide el proceso mediante validación cruzada. Sin embargo, aplicar esta recomendación toma una cantidad de tiempo considerable. Por lo tanto, la persona investigadora debe de considerar los tiempos de procesamiento computacional a la hora de realizar los análisis.

El análisis de sentimientos, dada la naturaleza de las preguntas analizadas, se centró en identificar los sentimientos negativos. Los léxicos generados manualmente son más precisos, sin embargo, tienden a presentar una cobertura a plazo relativamente bajo (Aminu, 2016). El número de palabras en el diccionario aumentará una vez que se estudien más documentos. Se podría alcanzar un estado estable una vez que se hayan analizado un gran número de documentos relacionados con la temática de transparencia.

El archivo de datos utilizado para realizar los análisis contiene tres variables de ponderación para ajustar los resultados a nivel nacional. Habría sido deseable ponderar los datos; sin embargo, esto no se hizo debido a la falta de recursos computacionales y la complejidad de las técnicas aplicadas en el documento.

El presente trabajo constituye una primera aproximación del uso de técnicas provenientes de la minería de texto para analizar la información contenida en preguntas abiertas en encuestas de opinión. Esta modalidad de análisis se debe seguir investigando y aplicando al contexto de Costa Rica, debido a que produce ventajas considerables. El análisis de este tipo de preguntas tradicionalmente se ha realizado mediante el método de la codificación manual; no obstante, las capacidades computacionales con las que se cuenta en la actualidad permiten realizar un análisis más automatizado de las preguntas abiertas.

BIBLIOGRAFÍA

Aggarwal, C y Zhai, C (2012). *Mining text data*. Springer. Estados Unidos. doi: http://doi.org/c38g.

Aguilera, R (2017). *La transparencia y la formación de ciudadanía en un gobierno local: Oportunidades y restricciones en Jalisco*. Universidad Nacional Autónoma de México. Estudios Políticos (43): 111-135.

Aigneren, M (2008). *Una propuesta de análisis de los datos*. Centros de Estudio de Opinión. Universidad de Antioquia. Colombia.

Alfaro, R., Cárdenas, J y Olivares, G (2014). *Clasificación automática de textos usando redes de palabras*. Revista Signos. doi: http://doi.org/gddj6n.

Alghamandi, R y Alfalqi, K (2015). *A survey of topic modelling in text mining*. International Journal of Advanced Computer Science and Applications 6(1).

Allahyari, M et al (2017). *A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques*.

Álvarez, R (2003). *Las preguntas de respuesta abierta y cerrada en los cuestionarios. Análisis estadístico de la información*. Universidad de León. España. Metodología de las encuestas (5): 45-54.

Aminu, M (2016). *Contextual Lexicon-based Sentiment Analysis for Social Media*. Tésis de doctorado. Universidad de Robert Gordon.

Ananiadou. S., Keek, D y Tsujii, J (2006). *Text mining and its potential applications in systems biology*. Science Direct. doi: http://doi.org/bd8pr8.

Araujon, N (2009). *Método Semisupervisado para la Clasificación Automática de Textos de Opinión.Puebla (Tesis de Maestría).* Instituto Nacional de Astrofísica, Óptica y Electrónica. México.

Armijo, F y Vives, J (2016). *La tutela jurídica en Costa Rica sobre el derecho de acceso a la información pública*. Facultad de Derecho. Universidad de Costa Rica.

Banea, C., Mihalcea, R., y Wiebe, J (2011). *Multilingual Sentiment and Subjectivity Analysis*. Multilingual Natural Language Processing.

Berry, M y kogan, J (2010). *Text Mining: Applications and Theory*. Wiley and Sons. Chichester. Reino Unido.

Breiman, L (2001). *Random forests, Machine Learning*. 45: 5–32.

Carmona, S. (2014). *Tutorial sobre Máquinas de Vectores Soporte (SVM)*. Departamento de Inteligencia Artificial. Universidad Nacional de Educación a Distancia. España.

Centeno, O (2016). *Encuesta Nacional de Percepción sobre la Transparencia 2016. Plan Estratégico Institucional 2013-2020*. Contraloría General de la República. División de Fiscalización Operativa y Evaluativa.

Consuelo, M (2017). *Nuevas técnicas de minería de textos: Aplicaciones*. Departamento de Ciencias de la Comunicación e Inteligencia Artificial. Universidad de Granada.

Contraloría General de la República (2019). *Memoria Anual 2018*. San José. Costa Rica.

Contreras, M (2014). *Minería de texto. Una visión actual*. Universidad Nacional Autónoma de México. Biblioteca Universitaria 2: 129-138.

Cutler, A., Cluter, D y Stevens, J (2011). Ensamble *Machine Learning. Random Forest.* doi: http://doi.org/c4xj

Dallas, C y Smith, N (2015). *Automated Coding of Open-Ended Survey Responses*.

Das, S. y Chen, M. (2001). *Yahoo! for Amazon: Extracting market sentiment from stock message boards*. Asia Pacific Finance Association Annual Conf.

Decreto Ejecutivo Nº 40200 MP-MEIC-MC (2017). *Transparencia y acceso a la información pública*. San José. Costa Rica.

Del Catillo, A (2003). *Medición de la corrupción: Un indicador de la Rendición de Cuentas*. Cultura de la rendición de cuentas. México.

Deng, X., Li, Y y Weng, J (2018). *Multimedia Tools Applications. Feature selection for text classification: A review*. Springer. doi: https://doi.org/10.1007/s11042-018-6083-5.

Eberendu, C (2016). Unstructured Data: An overview of the data of big data. Universidad de Madonna. doi: http://doi.org/dfq9.

Franc, V y Hlavac, V (2014). Multi-class Support Vector Machine. Proceedings 16th International Conference on Pattern Recognition (2), 236-239. Estados Unidos.

Fuchs, M y Bošnjak, M (2014). *Open-ended questions in Web surveys: using visual and adaptive questionnaire design to improve narrative responses*. Universidad de Darmstadt. Alemania.

Georgiou, D., MacFarlane, A y Russel, T (2015). *Extracting Sentiment from Healthcare Survey Data: An Evaluation of Sentiment Analysis Tools*. Science and Information Conference 2015. p. 352-361.

Giorgetti, D y Sebastiani, F (2003). *Automating survey coding by multiclass text categorization techniques*. doi: 10.1002/asi.10335.

Giorgetti, D., Prodanof, I y Sebastiani, F (2002). *Automated Coding of Open-ended Surveys: Technical and Ethical Issues*.

Godoy, A (2015). *Técnicas de aprendizaje de máquina utilizadas para la minería de texto*. a Universidad Federal de Santa Catarina. México. Investigación Bibliotecológica (31) 71.

Gurusamy, V y Kannan, S (2014). Preprocessing Techniques for Text Mining. Universidad de Madurai Kamaraj.

Hastie, T., Tibshirani, R y Friedman, J. (2001). *The Elements of Statistical Learning*. New York, NY, USA: Springer New York Inc.

Hearst, M (2003). *What is Text Mining?*. Universidad de Berkeley. Escuela de Gestión de la Información y Sistemas. Estados Unidos.

Hearst, M (1999). *Untangling Text Data Mining*. Universidad de Berkeley. Escuela de Gestión de la Información y Sistemas. Estados Unidos.

Hernández, O (2013). Temas de análisis estadístico multivariante. Editorial UCR.

Jain, A (2014). *Analyzing responses to open-ended questions for SPIRIT using aspects-oriented sentiment analysis*. Universidad Purdue.

Kareem, I y Duaimi, M (2014). *Improved Accuracy for Decision Tree Algorithm Based on Unsupervised Discretization*. International Journal of Computer Science and Mobile Computing 3. p. 176-183.

Kotelnikov, E., Bushmeleva, N., Razova, E., Peskisheva, T y Pletneva, M (2016). *Manually created sentiment lexicons: research and development*. Computational Linguistics and Intellectual Technologies. Rusia.

Kotzé, E (2018). *Employing sentiment analysis for gauging perceptions of minorities in multicultural societies: An analysis of Twitter feeds on the Afrikaner community of Orania in South Africa*. The Journal for Transdisciplinary Research in Southern Africa. doi: 10.4102/td.v14i1.564

Labille, K., Gauch, S y Alfarhood, S (2017). *Creating Domain Specific Sentiment Lexicons via Text Mining*. In Proceedings of Workshop on Issues of Sentiment Discovery and Opinion Mining. Canadá.

Lindstedt, C y Naurin, D (2005). *Transparency and Corruption. The Conditional Significance of a Free Press*. The QOG Institute Quality of Government. Goterborg University. Suecia.

Liu, B (2012). *Sentiment Analysis and Opinion Mining*. Morgan & Claypool.

Liu, L., Tang, L., Dong, W., Yao, S y Zhou, W (2016). *An overview of topic modelling and its current applications in bioinformatics*. doi: 10.1186/s40064-016-3252-8

Lord, K. M. (2006). *The perils and promise of global transparency*. Albany, NY: State University of New York Press.

Mabillard, V y Pasquier, M (2015). *Transparency and Trust in Government: Two-Way Relationship*. Université de Lausanne. Yearbook of Swiss Administrative Sciences: 23-34.

Madhusudanan, N., Gurumoorthy, B y Chakrabarti, A (2016). *Enhancing Domain Specific Sentiment Lexicon for Issue Identification*. doi: http://doi.org/c6kr. Springer. 13–21.

Maheswari1, M y Sathiaseelan, J (2015). *Text Mining: Survey on Techniques and Applications*. International Journal of Science and Research.

Mateo, J (2014). *Competición de Kaggle.com: Santander Customer Satisfaction (Tesis de Maestría).* Universidad Internacional de Andalucía. España.

Milios, E., Shafiei, M., Wang, S., Zhang, R., Tang, B y Tougas, J (2007). *A Systematic Study on Document Representation and Dimensionality Reduction for Text Clustering*. doi: http://doi.org/dm7cnd.

Mitchell, T (1997). *Machine Learning*. McGraw Hill. Nueva York. Estados Unidos.

Mladenić, D. (2011) *Feature Selection in Text Mining*. Encyclopedia of Machine Learning. Springer. Estados Unidos. doi: 978-0-387-30164-8.

Murtagh, F y Legendre, P (2011). *Ward's Hierarchical Clustering Method: Clustering Criterion and Agglomerative Algorithm.*

Neuman, M (2002). *Acceso a la información. La llave para la democracia*. El Centro Carter.

Nguyen, C., Rivero, J y Morell, C (2015). *Aprendizaje supervisado de funciones de distancia: estado del arte*. Revista Cubana de Ciencias Informáticas (9)2. 14-28.

Oliva, F (2014). *Minería de opinión y análisis de sentimientos*. Facultad de ingeniería. Pontificia Universidad Católica de Valparaíso. Chile.

Pang, B y Lee, L (2002). *Thumbs up? Sentiment Classification using Machine Learning Techniques*. In Proceedings of the Conference on Empirical Methods in Natural Language Processing. 79-86.

Pang, B y Lee, L (2008). *Opinion mining and sentiment analysis*. Foundations and Trends in Information Retrieval.

Pérez, M y Cardosa, C (2010). *Minería de texto para la categorización automática de documentos*. Universidad Católica de Salta. Argentina.

Pope, J (1981). *Investigación de Mercados*. Editorial Norma. Bogotá. Colombia.

Porumbescu, G y Im, T (2015). *Does transparency improve citizen's perceptions of government performance? Evidence from Seoul, South Korea.* SAGE Journal.

Redhu et al (2018). *Sentiment Analysis Using Text Mining: A Review*. International Journal on Data Science and Technology 4(2): 49-53. doi: http://doi.org/c3s2.

Reinsel, D., Gantz, J y Rydning, J (2018). *The digitization of the word. From edge to core*. International Data Corporation

Reja, U., Lozar, K., Hlebec, V y Vehovar, V (2003). *Open-ended vs. Close-ended Questions in Web Questionnaires*. Facultad de Ciencias Sociales. Universidad de Ljubljana. Eslovenia.

Relly, J y Sabharwal, M (2009). *Perceptions of transparency of government policymaking: A cross-national study*. Government Information Quarterly. Elsevier.

Rincón, G (2016) *Minería de textos y análisis de sentimientos en sanidadysalud.com*. Facultad de Estudios Estadísticos. Universidad Complutense de Madrid. España.

Rincón, W (2014). *Preguntas abiertas en encuestas ¿cómo realizar su análisis?.* Universidad de Santo Tomas. Colombia. Comunicaciones en Estadística 2: 139-156.

Rosas, O (2010). *La estructura disposicional de los sentimientos. Ideas y Valores*. Universidad de Twente. Holanda.

Sánchez, J (2015). *La participación ciudadana como instrumento del gobierno abierto*. Espacios Públicos18 (43). Universidad Autónoma del Estado de México. México.

Sebastiani, F. (2005). *Text categorization. In Text Mining and its Applications to Intelligence, CRM and Knowledge Management*. WIT Press. 109–129.

Serna, S. (2009). *Comparación de árboles de regresión y clasificación y regresión logística*. Universidad Nacional de Colombia. Escuela de Estadística. Colombia.

Shuster, J y Martínez, A (2014). *Transparencia y rendición de cuentas en el ámbito municipal*. Horizontes de la Contaduría (1). pp. 298-313.

Silge, J y Robinson, D (2019). *Text Mining with R. A Tidy Approach*. O'Reilly.

Sistema Costarricense de Información Jurídica. *Constitución Política de la Republica de Costa Rica*. Recuperado el 26 de marzo del 2019 de: http://www.pgrweb.go.cr/scij/Busqueda/Normativa/Normas/nrm_texto_completo.aspx?param1=NRTC&nValor1=1&nValor2=871&nValor3=0&strTipM=TC.

Solka, J (2008). *Statistics Surveys. Text Data Mining: Theory and Methods*. doi: 10.1214/07-SS016.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K y Stede, M (2011). *Lexicon-Based Methods for Sentiment Analysis.* Association for Computational Linguistics.

Tufféry, S (2011). *Data mining and statistics for decision making*. John Wiley & Sons.

Turney, P (2002*). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews*. Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics. pp. 417-424.

Vani, S., Appa, A., Sridhar, G., Chakravarthy, M., Nageshwararo, K y Rao, P (2010). Cluster analysis and phylogenetic relationship in biomarkers indentification of type 2 diabetes and nephropathy. doi: http://doi.org/fqfkcv

Viechnicki, P (1998*). A performance evaluation of automatic survey classifiers. Heidelberg*: Springer Verlag. Universidad de Chicago.

Xu, J., Liu, X., Huo, Z., Deng, C., Nie, F y Huang, H (2017). Multi-Class Support Vector Machine via Maximizing Multi-Class Margins. Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence. 3154-3160. doi: https://doi.org/10.24963/ijcai.2017/440.

Zaïane, O (1999). *Introduction to Data Mining*. Departamento de Ciencias de la Computación. Universidad de Alberta. Canada.

Zainuddin, N y Selamat, A (2014). *Sentiment Analysis Using Support Vector Machine*. International Conference on Computer, Communication, and Control Technology. Malasia. doi: 10.1109/I4CT.2014.6914200.

Zamora, D (2016). Informe de Resultados 2016. *Índice de Transparencia del Sector Público. Basado en sitios web.*

Zareapoor, M y Seeja, K (2015). *Feature Extraction or Feature Selection for Text Classification: A Case Study on Phishing Email Detection*. Information Engineering and Electronic Business 2. doi: http://doi.org/c6cj.

Züll, C (2016). *Open-Ended Questions. GESIS Survey Guidelines*. Institute for the Social Sciences. Alemania. doi: 10.15465/gesis-sg_en_002.

ANEXO

Anexo I. Cuadros referentes a los resultados de la predicción de respuestas

Cuadros pertenecientes al cuestionario de acceso a la información:

Cuadro 5. Comparación de clases codificadas manualmente con las clases predichas para la pregunta que hace referencia a impedimentos para obtener información en instituciones públicas

Codificación manual		Predicho		
Categoría	**Distribución porcentual (%)**	**Categoría**	**Distribución porcentual (%)**	**Precisión (%)**
Trámites	14,37	Trámites	17,39	72,25
Duración / Tiempo	12,26	Duración / Tiempo	12,06	82,50
Nada	10,75	Nada	11,26	86,61
Desconocimiento	9,35	Desconocimiento	9,65	61,46
Página web	8,04	Funcionarios públicos	8,84	68,18
Funcionarios públicos	7,84	Página web	8,64	75,58
No ha buscado info	6,53	No ha buscado info	7,44	82,43
Atención al cliente	5,63	Atención al cliente	5,73	63,16
Acceso a internet	4,52	Acceso a internet	4,12	85,37
Accesibilidad	2,71	Accesibilidad	2,61	57,69
Negativa de info	2,11	Horarios	1,81	83,33
Desactualización	1,91	Negativa de info	1,71	47,06
Lejanía a la institución	1,91	Desactualización	1,51	86,67
Horarios	1,81	Dificultad de obtenerla	1,31	53,85
Dificultad de obtenerla	1,61	Canales comunicación	1,31	38,46
Otra	1,61	Lejanía a la institución	1,21	66,67
Canales comunicación	1,51	No recibe respuesta	0,80	62,50
No recibe respuesta	1,51	Falta de claridad info	0,80	50,00
Falta de claridad info	1,11	Falta de interés	0,60	50,00
Falta de organización	1,11	Otra	0,50	40,00
Características de info	1,01	Falta de organización	0,50	0,00
Falta de interés	0,80	Características de info	0,20	0,00

Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

Cuadro 6. Comparación de clases codificadas manualmente con las clases predichas para la pregunta que hace referencia a sugerencias para eliminar la barrera

Codificación manual		Predicho		
Categoría	**Distribución porcentual (%)**	**Categoría**	**Distribución porcentual (%)**	**Precisión (%)**
Facilitar acceso	13,41	Facilitar acceso	15,76	57,85
Mejorar página web	10,29	Otro	15,23	29,91
Agilizar trámites y procesos	10,16	Mejorar página web	9,77	68,00
Capacitando	9,24	Agilizar trámites y procesos	7,94	80,33
Funcionarios públicos	8,98	Funcionarios públicos	7,55	74,14
Mejorar servicio al cliente	7,16	Capacitando	6,25	81,25
Otro	7,03	Mejorar servicio al cliente	5,60	51,16
Ampliar medios	3,91	Actualizar información	4,56	77,14
Actualizar información	3,78	Eficiencia	3,52	77,78
Eficiencia	3,78	Ampliar medios	3,26	52,00
Mejorar comunicación	2,86	Acceso a internet	2,08	62,50
Mayor control	1,95	Mejorar comunicación	1,82	64,29
Acceso a internet	1,69	Redes sociales	1,56	58,33
Normativa	1,69	Digitalizar información	1,30	80,00
Digitalizar información	1,56	Ampliar horario	1,30	70,00
Ampliar horario	1,17	Mayor control	1,17	66,67
Lugar para obtener info	1,17	Normativa	1,17	77,78
Mejorar info disponible	1,17	Más tecnología	1,17	44,44
Herramientas amigables	1,04	Transparencia	1,17	44,44
Redes sociales	1,04	Chat en línea	0,91	14,29
Más tecnología	0,78	Lugar para obtener Info	0,78	50,00
Organización institucional	0,78	Organización institucional	0,78	50,00
Combatir corrupción	0,65	Mejorar info disponible	0,65	40,00
Chat en línea	0,52	Combatir corrupción	0,65	80,00
Mayor disponibilidad	0,52	Herramientas amigables	0,52	25,00
Participación ciudadana	0,52	Respuesta a solicitudes	0,52	0,00
Sanciones a funcionarios	0,52	Mayor disponibilidad	0,39	100,00
Transparencia	0,52	Participación ciudadana	0,39	0,00
Denuncia	0,39	Sanciones a funcionarios	0,39	66,67
Respuesta a solicitudes	0,39	Contralorias de servicios	0,39	0,00
CGR	0,26	Rendicion de cuentas	0,39	0,00
Contralorias de servicios	0,26	Denuncia	0,26	100,00
Cumplimiento de plazos	0,26	CGR	0,26	0,00
Informarse	0,26	Cumplimiento de plazos	0,26	100,00
Rendición de cuentas	0,26	Informarse	0,26	50,00

Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

Cuadro 7. Comparación de clases codificadas manualmente con las clases predichas para la pregunta que hace referencia qué haría la persona desde su propio ámbito para solucionar la barrera

Codificación manual		Predicho		
Categoría	**Distribución porcentual (%)**	**Categoría**	**Distribución porcentual (%)**	**Precisión (%)**
Nada	13,83	Otro	14,15	30,68
Hacer sugerencias	11,25	Nada	13,67	88,24
Informarse	8,52	Hacer sugerencias	13,67	64,71
Otro	8,20	Informarse	8,84	65,45
Denunciar	5,95	Compartir información	5,47	44,12
Interés	4,66	Denunciar	5,14	93,75
Participación ciudadana	4,66	Interés	4,82	66,67
Insistir	3,86	Insistir	4,66	62,07
Capacitar	3,38	Participación ciudadana	3,38	80,95
Solicitar información	3,38	Solicitar información	3,22	55,00
Compartir información	3,22	Uso de internet	2,73	47,06
Mejorar comunicación	2,57	Capacitar	2,57	68,75
Quejarse	2,41	Quejarse	2,57	81,25
Uso de internet	2,41	Consultar a la institución	1,93	58,33
Consultar a la institución	2,09	Sacar tiempo	1,93	66,67
Control ciudadano	1,93	Mejorar comunicación	1,45	66,67
Opinar	1,61	Usar tecnología	1,45	44,44
Sacar tiempo	1,61	Simplificar trámites	1,29	50,00
Mejorar actitud	1,45	Paciencia	0,96	83,33
Simplificar trámites	1,45	Ley	0,80	100,00
Usar tecnología	1,29	Control ciudadano	0,64	100,00
Uso de páginas web	1,13	Mayor disponibilidad	0,64	75,00
Ley	0,96	Ser transparente	0,48	100,00
Comunidad organizada	0,80	Evaluar servicios	0,48	100,00
Paciencia	0,80	Elección de personal al poder	0,48	100,00
Proponer	0,80	Actualizar información	0,48	66,67
Ser eficiente	0,80	Mejorar actitud	0,32	50,00
Ser transparente	0,80	Uso de páginas web	0,32	0,00
Dar seguimiento	0,64	Comunidad organizada	0,32	50,00
Evaluar servicios	0,64	Proponer	0,32	50,00
Mayor disponibilidad	0,64	Ser eficiente	0,32	50,00
Elección de persononal al poder	0,48	Opinar	0,16	0,00
Facilitar acceso	0,48	Facilitar acceso	0,16	0,00
Hacer valer derechos	0,48	Hacer valer derechos	0,16	0,00
Usar recursos disponibles	0,48	Dar seguimiento	0,00	0,00
Actualizar información	0,32	Usar recursos disponibles	0,00	0,00

Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

Cuadros pertenecientes al cuestionario de rendición de cuentas:

Cuadro 8. Comparación de clases codificadas manualmente con las clases predichas para la pregunta que hace referencia a impedimentos para obtener resultados de lo que hace el sector público

Codificación manual		Predicho		
Categoría	**Distribución porcentual (%)**	**Categoría**	**Distribución porcentual (%)**	**Precisión (%)**
No se ha solicitado	20,83	No se ha solicitado	25,10	75,10
Falta de tiempo personal	11,56	Falta de tiempo personal	10,21	90,82
Accesibilidad	7,29	Falta información o claridad	7,60	47,95
Dificultad para encontrar la info	7,08	Sin impedimento	7,50	87,50
Sin impedimento	6,77	Dificultad para encontrar la info	6,98	73,13
Funcionario que atiende	6,56	Accesibilidad	6,88	84,85
Falta información o claridad	6,25	Funcionario que atiende	6,88	78,79
Burocracia	4,79	Burocracia	4,79	82,61
Desconocimiento	4,69	Desconocimiento	4,79	58,70
Página web	4,69	Página web	4,69	88,89
Falta de interés	4,06	Falta de interés	3,02	68,97
Trámites engorrosos	3,65	Tramites engorrosos	3,02	72,41
Desconfianza de la información	2,19	No se facilita	2,08	45,00
Información desactualizada	2,19	Ocultamiento de información	1,77	35,29
No se facilita	2,08	Información desactualizada	1,67	87,50
Ocultamiento de información	1,98	Desconfianza de la información	1,56	53,33
Otro	1,04	Disponibilidad	0,42	50,00
Disponibilidad	0,62	horario	0,42	100,00
No acceso por confidencialidad	0,62	Información falsa	0,31	100,00
Horario	0,52	No acceso por confidencialidad	0,21	50,00
Información falsa	0,52	Otro	0,10	0,00

Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

Cuadro 9. Comparación de clases codificadas manualmente con las clases predichas para la pregunta que hace referencia a sugerencias para solucionar la barrera

Codificación manual		Predicho		
Categoría	**Distribución porcentual (%)**	**Categoría**	**Distribución porcentual (%)**	**Precisión (%)**
Difusión	18,03	Difusión	31,67	43,54
Rediseño de la página	12,73	Rediseño de la página	12,27	82,72
Capacitar en servicio al cliente	9,39	Capacitar en servicio al cliente	8,64	77,19
Nuevos medios tecnológicos	8,64	Ampliar accesibilidad	6,97	54,35
Personal apto	6,21	Personal apto	6,21	51,22
Ampliar accesibilidad	4,70	Nuevos medios tecnológicos	4,55	50,00
Ciudadanos proactivos	3,94	Eficientizar	3,94	53,85
Eficientizar	3,94	Transparencia	3,64	62,50
Simplificar trámites	3,03	Ciudadanos proactivos	2,88	47,37
Agilidad	2,88	Actualización de la info	2,88	52,63
Información disponible	2,88	Incorporar más informacion	2,88	36,84
Mayor claridad	2,88	Agilidad	2,58	35,29
Transparencia	2,58	Simplificar trámites	2,42	87,50
Fiscalizar	2,42	Información disponible	1,67	45,45
Actualización de la info	2,27	Redes sociales	1,36	100,00
Incorporar más información	2,12	Mayor claridad	1,21	87,50
Oficina o persona encargada	1,97	Brindar la información	1,06	42,86
Otro	1,82	Oficina o persona encargada	0,76	100,00
Brindar la información	1,52	Otro	0,76	0,00
Redes sociales	1,52	Fiscalizar	0,45	66,67
Denunciar	1,36	Focalizarse en el ciudadano	0,45	0,00
Variedad accesos	1,06	Ninguna	0,30	100,00
Ninguna	0,61	Innovación	0,30	50,00
Normativa	0,61	Denunciar	0,15	100,00
Focalizarse en el ciudadano	0,45	Variedad accesos	0,00	0,00
Innovación	0,45	Normativa	0,00	0,00

Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

Cuadro 10. Comparación de clases codificadas manualmente con las clases predichas para la pregunta que hace referencia a qué haría la persona desde su propio ámbito para solucionar la barrera

Codificación manual		Predicho		
Categoría	**Distribución porcentual (%)**	**Categoría**	**Distribución porcentual (%)**	**Precisión (%)**
Exigir derechos	18,03	Exigir derechos	28,05	50,60
Nada	13,69	Informarse	14,02	75,00
Informarse	13,36	Nada	12,85	89,61
Mayor interés	13,36	Mayor interés	10,52	76,19
Sugerencias	6,51	Sugerencias	6,01	94,44
Ciudadanos proactivos	6,01	Denunciar	6,01	80,56
Denunciar	5,68	Compartir información	4,84	37,93
Fiscalizar	3,17	Ciudadanos proactivos	4,01	70,83
Estar atento a la información	3,01	Presentar queja	2,67	75,00
Presentar queja	3,01	Fiscalizar	2,34	35,71
Compartir información	2,67	Estar atento a la información	1,84	54,55
Agruparse	2,00	Dar ejemplo	1,84	9,09
Dar ejemplo	1,84	Fortalecer comunicación	1,34	87,50
Participación activa	1,84	Participación activa	1,17	71,43
Fortalecer comunicación	1,50	Colaborar con el gobierno	1,00	33,33
Muy poco	1,50	Muy poco	0,83	80,00
Colaborar con el gobierno	1,34	Ser más consientes	0,33	0,00
Ser más consientes	0,83	Agruparse	0,17	0,00
Capacitarse	0,67	Capacitarse	0,17	100,00

Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

Cuadros pertenecientes al cuestionario de participación ciudadana:

Cuadro 11. Comparación de clases codificadas manualmente con las clases predichas para la pregunta que hace referencia a sugerencias para eliminar la barrera

Codificación manual		Predicho		
Categoría	**Distribución porcentual (%)**	**Categoría**	**Distribución porcentual (%)**	**Precisión (%)**
Mejorar comunicación	20,06	Mejorar Comunicación	29,59	48,34
Horarios flexibles	15,99	Horarios flexibles	15,85	90,27
Divulgar información	14,45	Divulgar información	14,59	71,15
Uso de tecnología	7,99	Uso de tecnología	7,57	72,22
Abrir espacios de participación	4,77	Tomar en cuenta al ciudadano	4,49	50,00
Capacitar	4,77	Abrir espacios de participación	4,21	50,00
Otro	4,49	Otro	3,93	25,00
Tomar en cuenta al ciudadano	4,07	Trabajar en accesibilidad	3,51	68,00
Trabajar en accesibilidad	3,23	Capacitar	3,23	73,91
Agilizar procesos y trámites	2,81	Agilizar procesos y trámites	2,81	70,00
Funcionarios	2,66	Promover cultura participativa	1,82	38,46
Promover cultura participativa	2,38	Cambiar metodologías	1,40	70,00
Cambiar metodologías	1,82	Funcionarios	1,12	62,50
Cultura participativa	1,26	Prevenir corrupción	1,12	100,00
Presencia en comunidades	1,26	Cultura participativa	0,98	42,86
Prevenir corrupción	1,12	Rendición de cuentas	0,56	0,00
Rendición de cuentas	0,84	Transparencia	0,56	75,00
Transparencia	0,84	Mayor interés	0,56	50,00
Cumplir ley	0,70	Presencia en comunidades	0,42	66,67
Grupos organizados	0,70	Acercarse a las comunidades	0,42	100,00
Permisos laborales	0,70	Cumplir ley	0,28	50,00
Atención al ciudadano	0,56	Permisos laborales	0,28	100,00
Control ciudadano	0,56	Control ciudadano	0,28	50,00
Denunciar	0,56	Grupos organizados	0,14	0,00
Acercarse a las comunidades	0,42	Atención al ciudadano	0,14	0,00
Mayor interés	0,42	Eficiencia	0,14	0,00
Eficiencia	0,28	Denunciar	0,00	0,00
Encuestas	0,28	Encuestas	0,00	0,00

Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

Cuadro 12. Comparación de clases codificadas manualmente con las clases predichas para la pregunta que hace referencia a qué haría la persona desde su propio ámbito para solucionar la barrera

Codificación manual		Predicho		
Categoría	**Distribución porcentual (%)**	**Categoría**	**Distribución porcentual (%)**	**Precisión (%)**
Involucrarse	33,28	Involucrarse	41,28	67,61
Informarse	15,55	Informarse	15,99	64,55
Opinar	10,90	Opinar	10,76	72,97
Nada	7,41	Compartir información	7,41	56,86
Compartir información	6,69	Nada	6,98	93,75
Otro	4,80	Otro	3,63	40,00
Apoyar	3,63	Apoyar	2,47	47,06
Crear espacios de participación	2,18	Denunciar	2,33	50,00
Denunciar	2,18	Medios electrónicos	1,60	72,73
Medios electrónicos	2,18	Grupos organizados	1,45	50,00
Grupos organizados	1,60	Crear espacios de participación	1,02	28,57
Insistir	1,31	Quejarse	0,87	100,00
Control ciudadano	1,02	Horarios flexibles	0,73	40,00
Ser consciente	1,02	Insistir	0,58	50,00
Investigar	0,87	Control ciudadano	0,58	0,00
Pedir cambio de horarios	0,87	Investigar	0,58	100,00
Quejarse	0,87	Acercarse a la institución	0,44	66,67
Comunicarse con vecinos	0,73	Ser consciente	0,29	50,00
Proponer	0,73	Comunicarse con vecinos	0,29	50,00
Acercarse a la institución	0,58	Promover participación	0,29	0,00
Capacitar	0,44	Pedir cambio de horarios	0,15	0,00
Horarios flexibles	0,44	Proponer	0,15	0,00
Promover participación	0,44	Ejercer derecho	0,15	0,00
Ejercer derecho	0,29	Capacitar	0,00	0,00

Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

Cuadros pertenecientes al cuestionario dirigido a funcionarios públicos:

Cuadro 13. Comparación de clases codificadas manualmente con las clases predichas para la pregunta que hace referencia a la principal barrera que le impide a la institución en la que labora el funcionario público ser más transparente ante la ciudadanía

Codificación manual		Predicho		
Categoría	**Distribución porcentual (%)**	**Categoría**	**Distribución porcentual (%)**	**Precisión (%)**
Ninguna	14,21	Ninguna	16,91	78,72
Aspectos políticos	12,23	Burocracia	12,95	87,50
Burocracia	12,05	Aspectos políticos	12,77	64,79
Deficiencias en comunicación	8,27	Deficiencias de accesibilidad a la información	8,09	53,33
Deficiencias de accesibilidad a la Información	7,73	Deficiencias en comunicación	7,55	59,52
Actitud de funcionarios	6,65	Desconocimiento	6,83	47,37
Falta de recursos	6,47	Falta de recursos	6,65	43,24
Desconocimiento	5,94	Actitud de funcionarios	5,76	53,12
Aspectos normativos - legales	4,68	Barreras tecnológicas	3,96	40,91
Corrupción	4,32	Aspectos normativos - legales	3,78	66,67
Barreras tecnológicas	3,60	Corrupción	3,78	66,67
Deficiencias en estructura del estado	3,60	Deficiencias en estructura del estado	3,06	35,29
Manejo información confidencial o sensible	2,70	Manejo información confidencial o sensible	3,06	58,82
Mala administración de recursos	2,34	Otra	1,62	44,44
Otra	1,98	Mala administración de recursos	1,26	14,29
Deficiencias en procedimientos	1,08	Información muy técnica	0,72	100,00
Informacion muy técnica	1,08	Favoritismos	0,54	66,67
Favoritismos	0,72	Deficiencias en procedimientos	0,36	50,00
Manipulación de información	0,36	Manipulacion de información	0,36	0,00

Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

Cuadro 14. Comparación de clases codificadas manualmente con las clases predichas para la pregunta que hace referencia a: si el funcionario público entrevistado fuera jerarca, ¿qué haría para eliminar las barreras que le impiden a la institución ser más transparente

Codificación manual		Predicho		
Categoría	**Distribución porcentual (%)**	**Categoría**	**Distribución porcentual (%)**	**Precisión (%)**
Mejoras en proceso administrativo	13,10	Mejoras en proceso administrativo	18,56	42,35
Mejorar canales de comunicación	10,92	Mejorar presentación de información	12,23	48,21
Mejorar presentación de Información	10,26	Mejorar canales de comunicación	11,79	55,56
Agilizar procesos	8,95	Agilizar procesos	10,04	63,04
Capacitar a sus colaboradores	7,86	Capacitar a sus colaboradores	9,83	60,00
Mejorar transparencia	6,99	Mejorar transparencia	9,17	42,86
Mejorar capacidades tecnológicas	5,90	Mejorar capacidades tecnológicas	5,46	52,00
Mejorar normativa	5,68	Mejorar normativa	5,02	56,52
Acercarse más a los usuarios	5,02	Acercarse más a los usuarios	3,71	70,59
Otros	4,37	Mejorar controles	3,06	28,57
Mejorar controles	3,93	Otros	2,84	0,00
Brindar más apoyo a usuarios	3,71	Brindar más apoyo a usuarios	2,40	18,18
Cumplir marco normativo	2,62	Cumplir marco normativo	2,18	40,00
Digitalizar información	1,97	Digitalizar información	1,09	80,00
Actuar con más independencia	1,53	Actuar con más independencia	0,87	50,00
Revisar mecanismos actuales	1,53	Actualizar información	0,66	33,33
Actualizar información	1,31	Eliminar prácticas irregulares	0,66	0,00
Aplicar política de datos abiertos	1,31	Revisar mecanismos actuales	0,44	50,00
Eliminar prácticas irregulares	1,31	Aplicar política de datos abiertos	0,00	0,00
Capacitarse	0,66	Capacitarse	0,00	0,00
Sancionar funcionarios que incumplan	0,66	Sancionar funcionarios que incumplan	0,00	0,00
Aclarar mecanismos para solicitar Información	0,44	Aclarar mecanismos para solicitar información	0,00	0,00

Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

Cuadro 15. Comparación de clases codificadas manualmente con las clases predichas para la pregunta que hace referencia a qué haría la persona desde su propio ámbito para solucionar la barrera

Codificación manual		Predicho		
Categoría	**Distribución porcentual (%)**	**Categoría**	**Distribución porcentual (%)**	**Precisión (%)**
Mejorar presentación de información	16,93	Mejorar forma de trabajar	32,72	30,77
Nada	15,56	Mejorar presentación de información	24,94	55,96
Mejorar forma de trabajar	14,87	Nada	13,50	93,22
Aportar desde su puesto	9,84	Ser más transparente	5,49	75,00
Ser más transparente	8,01	Mejorar atención al público	5,03	72,73
Mejorar atención al público	7,09	Aportar desde su puesto	4,81	42,86
Capacitandose	4,81	Denunciar Irregularidades	2,97	69,23
Otra	4,58	Acercarse más al ciudadano	2,52	36,36
Acercarse más al ciudadano	4,35	Capacitandose	1,83	62,50
Agilizar procesos	3,66	Otra	1,60	14,29
Denunciar irregularidades	3,66	Agilizar procesos	1,60	85,71
Cumplir normativa	2,29	Cumplir normativa	1,60	28,57
Mejorar normativa	1,37	Digitalizar	0,69	100,00
Trabajar en función de Necesidades ciudadanas	1,14	Mejorar normativa	0,23	0,00
Capacitar colaboradores	0,92	Trabajar en función de Necesidades ciudadanas	0,23	100,00
Digitalizar	0,92	Capacitar colaboradores	0,23	0,00

Fuente: Contraloría General de la República. Encuesta Nacional de Percepción sobre la Transparencia 2019.

Anexo II. Lexicón

Cuadro 16. Lexicón creado referente a barreras para participar, obtener y acceder a información relacionada a asuntos del sector público

Palabra	Puntaje	Palabra	Puntaje	Palabra	Puntaje	Palabra	Puntaje	Palabra	Puntaje
negación	6	extranjera	2	apatía	1	escond	1	obstaculizan	1
negado	6	extranjero	2	argolla	1	escueto	1	obstáculo	1
negar	6	funcionaria	2	atención	1	espera	1	omit	1
negativa	6	funcionario	2	atiend	1	esperar	1	omitir	1
negativo	6	hermetismo	2	atraso	1	excesivo	1	paso	1
niegan	6	incompleta	2	barrera	1	exceso	1	pereza	1
nieguen	6	ineficiencia	2	brecha	1	excluyen	1	personalment	1
censura	5	ineficient	2	canal	1	extenso	1	poca	1
censurar	5	inoperante	2	cansancio	1	falta	1	poco	1
confidenci	5	insuficient	2	centralizada	1	faltar	1	polarización	1
corrupción	5	jefatura	2	cobertura	1	favoritismo	1	ponen	1
confidencial	4	limitada	2	complejo	1	fila	1	privar	1
confidencialidad	4	mal	2	complicado	1	gerencia	1	problema	1
esconden	4	mala	2	complicar	1	gubernament	1	proceso	1
desactualización	3	malo	2	comunicación	1	horario	1	prolongar	1
desactualizada	3	nicaragüens	2	confuso	1	ignorancia	1	protocolo	1
desactualizados	3	no_ineficient	2	contradiccion	1	ignorant	1	reglamento	1
falsa	3	ocultar	2	cuesta	1	inacceso	1	ridículo	1
grosero	3	ocultarla	2	cultura	1	incapacidad	1	salud	1
ignoran	3	página	2	desconoc	1	incompleta	1	saturada	1
ignorar	3	papel	2	desconocimiento	1	inconvenient	1	saturado	1
ley	3	papeleo	2	descontento	1	indisposición	1	saturarada	1
negligencia	3	papelería	2	desigu	1	inoperant	1	sistema	1
negligent	3	plataforma	2	desigualdad	1	internet	1	tardan	1
pésimo	3	política	2	desinformación	1	laboral	1	tardío	1
rechazo	3	político	2	desinformado	1	largo	1	técnica	1
restringen	3	politiquería	2	desinteré	1	lejanía	1	tecnicismo	1
abuso	2	religión	2	desorden	1	lejo	1	técnico	1
acceso	2	requisito	2	difícil	1	lenta	1	tedioso	1
accesoibilidad	2	restriccion	2	dificultad	1	lentitud	1	teléfono	1
accesoibilidad	2	servicio	2	dispon	1	lento	1	teme	1
actualizar	2	tergiversada	2	disponibilidad	1	lerda	1	temor	1
burocracia	2	trámite	2	distancia	1	lerdo	1	tiempo	1
desconfianza	2	trámitear	2	divulgación	1	limitacion	1	traba	1
difusa	2	trámiteo	2	duración	1	lugar	1	trabajo	1
difuso	2	trato	2	duran	1	manipular	1	trabamiento	1
distorsionada	2	xenofobia	2	edad	1	medio	1	ubicación	1
engorrosa	2	jerarca	2	enredo	1	miedo	1	ubicar	1
engorroso	2	jerarquía	2	error	1	monotonía	1	web	1
escasez	2	politizan	2	escasa	1	no	1	normativa	1
escueta	2	actitud	1	escepticismo	1	obsoleto	1	regulacion	1

Anexo III. Código de R

```
### Cargando librerias

suppressMessages(library(haven))
suppressMessages(library(tm))
suppressMessages(library(tidytext))
suppressMessages(library(tidyverse))
suppressMessages(library(SnowballC))
suppressMessages(library(hunspell))
suppressMessages(library(stringr))
suppressMessages(library(caret))
suppressMessages(library(rJava))
suppressMessages(library(RWeka))
suppressMessages(library(wordcloud))
suppressMessages(library(dplyr))
suppressMessages(library(ggplot2))
suppressMessages(library(plotly))
suppressMessages(library(viridisLite))
suppressMessages(library(RColorBrewer))
suppressMessages(library(e1071))
suppressMessages(library(rpart))
suppressMessages(library(randomForest))
suppressMessages(library(kknn))
suppressMessages(library(xgboost))
suppressMessages(library(tictoc))
suppressMessages(library(knitr))
suppressMessages(library(kableExtra))
suppressMessages(library(formattable))
suppressMessages(library(ggraph))
suppressMessages(library(igraph))
suppressMessages(library(visNetwork))
suppressMessages(library(widyr))
suppressMessages(library(factoextra))
suppressMessages(library(ape))
suppressMessages(library(GGally))
suppressMessages(library(ggpubr))
suppressMessages(library(grid))
suppressMessages(library(gridExtra))
```

```
### Función para corregir ortografía con Hunspell

corrector.ortografia <- function(datos_texto, diccionario)
  {
```

```
 datos_corregidos <- data.frame(texto = datos_texto)
 datos_corregidos$texto <- as.character(datos_corregidos$texto)
 datos_corregidos$nuevo <- 0

  for (i in 1:dim(datos_corregidos)[1])
   {

   respuesta <- datos_corregidos[i,"texto"]
   incorrectas <- hunspell(respuesta, dict = diccionario)
   buenas <- NULL

    if(length(unlist(incorrectas)) > 0)
     {

     for (k in 1:length(incorrectas[[1]]))
      {

       buenas.sugeridas <- hunspell_suggest(incorrectas[[1]][k], dict = diccionario)
       buenas[k] <- buenas.sugeridas[[1]][1]
      }

    names(buenas) <- unlist(incorrectas)
    datos_corregidos$nuevo[i] <- str_replace_all(respuesta, buenas)

     }
      else
         {

          datos_corregidos$nuevo[i] <- datos_corregidos[i,"texto"]

         }

   }

  return(datos_corregidos)

}
```

Función para lematizar los datos

```
corrector.lematizador <- function(datos_corregidos, diccionario)
 {

 datos_corregidos$nuevo <- as.character(datos_corregidos$nuevo)
 datos_corregidos$nuevo_stem <- 0
```

```
  for (i in 1:dim(datos_corregidos)[1])
   {

   respuesta <- datos_corregidos[i,"nuevo"]
   palabras <- hunspell_parse(respuesta, format = "latex")
   proveniente <- hunspell_stem(unlist(palabras), dict = diccionario)

   texto <- NULL

     if(length(unlist(proveniente)) > 0)
      {

      for (j in 1:length(proveniente))
       {

        palabra <- proveniente[[j]][1]
        texto <- paste(texto, palabra, sep = " ")
        }

      datos_corregidos$nuevo_stem[i] <- texto
      }
       else
        {

      datos_corregidos$nuevo_stem[i] <- respuesta

        }

    }

  return(datos_corregidos)
}
```

```
### Función para limpiar texto
```

```
limpiar.cuerpo <- function(datos)
 {

 corpus <- VCorpus(VectorSource(datos$nuevo_stem))
 corpus <- tm_map(corpus, removeNumbers)
 corpus <- tm_map(corpus, removePunctuation)
 corpus <- tm_map(corpus, content_transformer(tolower))
 corpus <- tm_map(corpus, stripWhitespace)
 corpus <- tm_map(corpus, stemDocument, language="spanish")
 my_stopwords <- setdiff(stopwords("spanish"), c("nada", "poco"))
 corpus <- tm_map(corpus, removeWords, my_stopwords)
```

```
  corpus <- tm_map(corpus, removeWords, c("dond","dónde", "si"))

  return(corpus)

}

limpiar.cuerpo.sentimientos <- function(datos)
  {

  corpus <- VCorpus(VectorSource(datos$nuevo_stem))
  corpus <- tm_map(corpus, removeNumbers)
  corpus <- tm_map(corpus, removePunctuation)
  corpus <- tm_map(corpus, content_transformer(tolower))
  corpus <- tm_map(corpus, stripWhitespace)
  corpus <- tm_map(corpus, stemDocument, language="spanish")
  my_stopwords <- setdiff(stopwords("spanish"), c("nada", "poco", "no"))
  corpus <- tm_map(corpus, removeWords, my_stopwords)
  corpus <- tm_map(corpus, removeWords, c("dond","dónde", "más", "cómo"))
  corpus <- tm_map(corpus, stripWhitespace)

  return(corpus)

  }
```

Función para crear la Matriz Documento-Palabra

```
MatrizDocumentoPalabra <- function(cuerpo)
  {

  tdmtrain <- DocumentTermMatrix(cuerpo
                        #control = list(tokenize = BigramTokenizer)
                         )
  datos <- as.matrix(tdmtrain)
  datos <- data.frame(datos)
  return(datos)

  }
```

Función para crear la Matriz Palabra-Documento

```
MatrizPalabraDocumento <- function(cuerpo)
  {

  tdmtrain <- TermDocumentMatrix(cuerpo
                        #control = list(tokenize = BigramTokenizer)
```

```
                        )
  datos <- as.matrix(tdmtrain)
  datos <- data.frame(datos)
  return(datos)

  }
```

Función para minúsculas

```
minuscula <- function(datos)
  {

  datos$nuevo <- str_to_lower(datos$nuevo, locale = "spanish")
  datos$nuevo <- str_replace_all(datos$nuevo, "paginas", "página")
  datos$nuevo <- str_replace_all(datos$nuevo, "pagina", "página")

  return(datos)

  }
```

Función para verificar datos finales

```
verificar.datos.finales <- function(datos, cuerpo)
  {

  datos$final_limpio <- 0

    for (k in 1:dim(datos)[1])
      {

       datos$final_limpio[k] <- cuerpo[[k]][1]
      }

  return(datos)

}
```

Función para hacer nube con unigramas

```
nube.unigrama <- function(cuerpo, titulo)
  {

  termdoc <- TermDocumentMatrix(cuerpo)
  termdoc <- as.matrix(termdoc)

  #freq.palabras <- termdoc %>%
```

```
 #              rowSums()

 v <- sort(rowSums(termdoc),decreasing=TRUE)
 d <- data.frame(word = names(v),freq=v)


  wordcloud(words = d$word,
          freq = d$freq,
          min.freq = 1,
          max.words = 120,
          random.order = FALSE,
          colors = brewer.pal(5, "Set1"))

  title(main = titulo)

 }
```

Función para hacer nube con bigramas

```
nube.bigrama <- function(cuerpo)
  {

  BigramTokenizer <- function(x) NGramTokenizer(x, Weka_control(min = 2, max = 2))

  tdm.bigram = TermDocumentMatrix(cuerpo,
                          control = list(tokenize = BigramTokenizer))

  freq = sort(rowSums(as.matrix(tdm.bigram)),decreasing = TRUE)
  freq.df = data.frame(word=names(freq), freq=freq)

  return(
  wordcloud(freq.df$word,freq.df$freq,
          max.words = 40,
          random.order = F,
          colors = brewer.pal(5, "Set1"),
          random.color = TRUE,
          ordered.colors = FALSE)
       )
#title(main = titulo)

 }
```

Función para obtener correlaciones en el texto

```
correlaciones.top10 <- function(cuerpo)
  {
```

```
  frequency <- as.matrix(
                    TermDocumentMatrix(
                                cuerpo
                               )
                   ) %>%
              rowSums() %>%
              sort(decreasing = TRUE)

  frequency.top10 <- frequency[1:10]

   for (i in 1:length(frequency.top10))
     {

      Correl <- findAssocs(TermDocumentMatrix(cuerpo),
                    names(frequency.top10[i]),
                    0.01)

      print(paste("Palabras asociadas con", names(frequency.top10[i])))
      print(Correl[[1]][1:10])

     }

}
```

Función para convertir datos a tipo numérico

```
convertidor.numerico <- function(datos)
 {

  for (k in 1:dim(datos)[2])
    {

     datos[ , k] <- as.numeric(as.character(datos[ , k]))

    }

  return(datos)

 }
```

Función para convertir datos a tipo factor

```
convertidor.factor <- function(datos)
 {
```

```
  for (k in 1:dim(datos)[2])
    {

   datos[ , k] <- as.factor(datos[ , k])

    }

  return(datos)
 }
```

Función para eliminar que se repiten solo una vez

```
eliminando.palabras <- function(datos)
  {

   num.col <- NULL
    for (k in 1:dim(datos)[2])
      {

       sum <- sum(datos[ , k])

         if(sum == 1)
           {

           num.col[k] <- k
           }
      }

   datos[ , as.vector(na.omit(num.col))] <- NULL

   print(paste("El total de palabras que quedan como predictoras son",
             dim(datos)[2]-1), sep = ",")

   return(datos)

 }
```

Función para hacer análisis de sentimiento

```
sentimientos <- function(respuestas, lexicon)
  {

  ##La variable con el texto se debe de llamar texto, mientras que en el lexicon la variable con los
puntajes se debe de llamar puntajes

  respuestas$texto <- apply(data.frame(respuestas$texto), 1,
```

```
                    function(x) encargarse.no(x))


respuestas <- tibble::rowid_to_column(respuestas, "id")

colnames(respuestas)[1] <- "linea"

tidy <- respuestas %>% unnest_tokens(palabra, texto)

merged <- merge(tidy,lexicon, by = 'palabra')

merged$puntajes <- as.numeric(merged$puntajes)

puntajes.finales <- aggregate(cbind(puntajes) ~ linea, data = merged,
                    FUN = sum)

respuestas.puntaje <- merge(respuestas, puntajes.finales, by ='linea')
##Agregar los que son neutrales
respuestas$puntajes <- 0

diff <- setdiff(respuestas$linea, respuestas.puntaje$linea)


respuestas.10 <- respuestas[diff, ]

respuestas.final <- rbind(respuestas.puntaje, respuestas.10)
respuestas.final$puntajes[respuestas.final$puntajes < 0] <- 0

respuestas.final$linea <- NULL

return(respuestas.final)

}
```

Función para encargarse de la negación

```
encargarse.no <- function(x)
 {

 str_split <- unlist(strsplit(x = x, split = " "))

 es_negativo <- grepl("no" , str_split, ignore.case = T)

 negar <- append(FALSE, es_negativo)[1:length(str_split)]

 str_split[negar == T] <- paste0("no_", str_split[negar == T])
```

```
  as.character(paste(str_split, collapse = " "))

  }
```

Función para calibración de XGBoost

```
XGBoost.calibracion <- function(datos)
  {

  xgbTrControl <- trainControl(
                        method = "repeatedcv",
                        number = 10,
                        repeats = 1,
                        summaryFunction = multiClassSummary)


  grid = expand.grid(
                nrounds = c(70, 100, 200),
                lambda = c(0),
                alpha = c(0),
                eta = c(0.1, 0.2, 0.3))


  MejorCalibracion <- train(var_respuesta ~ .,
                        data = datos,
                        method = "xgbLinear",
                        tuneGrid = grid,
                        trControl = xgbTrControl,
                        verbose = TRUE)

  return(MejorCalibracion)

 }
```

Función para calibración de Bosques aleatorios

```
Bosques.calibracion <- function(datos)
  {

  rfTrControl <- trainControl(
                        method = "repeatedcv",
                        number = 10,
                        repeats = 1,
                        summaryFunction = multiClassSummary)
```

```
 grid = expand.grid(
              mtry = c(100, 200, 300)
              )
 suppressWarnings(
 MejorCalibracion <- train(var_respuesta ~ .,
                     data = datos,
                     method = "rf",
                     tuneGrid = grid,
                     trControl = rfTrControl)
             )

 return(MejorCalibracion)

 }
```

Función para seleccionar variables con Ji-cuadrado

```
chi.cuadrado <- function(datos, umbral, respuesta = "var_respuesta")
 {

  num.col <- NULL
   for (k in 1:dim(datos)[2])
    {

     #datos[ , k] <- as.numeric(unlist(datos[ , k]))

     chi <- chisq.test(table(datos[ , respuesta], datos[ , k]),
                  simulate.p.value = TRUE)

       if(chi$p.value >= umbral)
         {

         num.col[k] <- k
         }
    }

  datos[ , as.vector(na.omit(num.col))] <- NULL

  print(paste("El total de palabras que quedan como predictoras después de eliminar las que tiene
un p-value mayor a", umbral, "son",
            dim(datos)[2]-1), sep = ",")

  return(datos)

}
```

```
### Función para entrenar modelos de clasificación de texto, con validación cruzada

modelos.clasificacion.texto <- function(datos,
                                XGBoost.calibracion,
                                Bosques.calibracion,
                                cv.folds,
                                cv.repetir)
 {

 datos$var_respuesta <- as.factor(datos$var_respuesta)

 n <- dim(datos)[1]

 ## k vecinos optimos para el algoritmo de vecinos más cercanos
 k.optimo <- floor(sqrt(nrow(datos)))

 errorsvmlini.CV <- 0
 errorsvmpoli.CV <- 0
 errorsvmradi.CV <- 0
 errorsvmsigi.CV <- 0
 errorRFi.CV <- 0
 errorBayesi.CV <- 0
 errorknni.CV <- 0
 errorxgboosti.CV <- 0

 datos <- tibble::rowid_to_column(datos, "id")

 ResultadoVerdadero.svmlin.CV <- data.frame(id = datos$id, resp_verdadera =
datos$var_respuesta)
 ResultadoVerdadero.svmpol.CV <- data.frame(id = datos$id, resp_verdadera =
datos$var_respuesta)
 ResultadoVerdadero.svmrad.CV <- data.frame(id = datos$id, resp_verdadera =
datos$var_respuesta)
 ResultadoVerdadero.svmsig.CV <- data.frame(id = datos$id, resp_verdadera =
datos$var_respuesta)
 ResultadoVerdadero.rf.CV <- data.frame(id = datos$id, resp_verdadera = datos$var_respuesta)
 ResultadoVerdadero.Bayes.CV <- data.frame(id = datos$id, resp_verdadera =
datos$var_respuesta)
 ResultadoVerdadero.knn.CV <- data.frame(id = datos$id, resp_verdadera = datos$var_respuesta)
 ResultadoVerdadero.xgboost.CV <- data.frame(id = datos$id, resp_verdadera =
datos$var_respuesta)

 #### Hiperparámetros de XGBoost

 parametrosXGBoost <- list(booster = "gbtree",
```

```
                    objective = "multi:softprob",
                    eval_metric = "merror",
                    eta = as.numeric(XGBoost.calibracion$bestTune[4]),
                    gamma = 0,
                    max_depth = 20,
                    min_child_weight = 1,
                    subsample = 0.90,
                    colsample_bytree = 0.90)

for (i in 1:cv.repetir)
  {

    errorsvmlini <- 0
    #errorsvmpoli <- 0
    errorsvmradi <- 0
    errorsvmsigi <- 0
    errorRFi <- 0
    errorBayesi <- 0
    errorknni <- 0
    errorxgboosti <- 0

    new_catsvmlin <- NULL
    #new_catsvmpol <- NULL
    new_catsvmrad <- NULL
    new_catsvmsig <- NULL
    new_catrf <- NULL
    new_catBayes <- NULL
    new_catknn <- NULL
    new_catxgboost <- NULL

    new_catsvmlin.ALL <- NULL
    #new_catsvmpol.ALL <- NULL
    new_catsvmrad.ALL <- NULL
    new_catsvmsig.ALL <- NULL
    new_catrf.ALL <- NULL
    new_catBayes.ALL <- NULL
    new_catknn.ALL <- NULL
    new_catxgboost.ALL <- NULL

    print(paste("----Está en la repetición----", i), sep = " ")

    ## Validación cruzada
    #tic()

    grupo <- createFolds(1:n, cv.folds)
```

```
for(j in 1:cv.folds)
  {

    muestra <- grupo[[j]]

    ttest <- datos[muestra, ]
    train <- datos[-muestra, ]

    train.1 <- train
    train <- eliminando.palabras.training(
                                  dplyr::select(train.1,
                                          -var_respuesta)
                                 )

    train$var_respuesta <- train.1$var_respuesta

    ttest <- ttest[ , colnames(train)]


    ttest.factor <- convertidor.factor(ttest)
    train.factor <- convertidor.factor(train)

    niveles <- levels(as.factor(train$var_respuesta))

    train$id <- NULL
    TestId <- ttest$id
    ttest$id <- NULL

    train$var_respuesta <- droplevels(train$var_respuesta)
    ttest$var_respuesta <- droplevels(ttest$var_respuesta)


    ## SVM Núcleo linear

    modsvmlin <- svm(var_respuesta ~ ., data = train,
                kernel = "linear",
                type = "C-classification")


    predictionsvmlin <- predict(modsvmlin, ttest)

    MC <- table(factor(ttest$var_respuesta, levels = niveles, ordered = TRUE),
                  factor(predictionsvmlin, levels = niveles, ordered = TRUE))
    acierto <- sum(diag(MC))/sum(MC)
```

```
error <- 1 - acierto
errorsvmlini <- errorsvmlini + error

new_catsvmlin <- data.frame(id = TestId, predictionsvmlin)
new_catsvmlin.ALL <- rbind(new_catsvmlin.ALL, new_catsvmlin)


## SVM Núcleo polinomial

#modsvmpol <- svm(var_respuesta ~ ., data = train,
 #            kernel = "polynomial",
  #           type = "C-classification")


#predictionsvmpol <- predict(modsvmpol, ttest)

#MC <- table(factor(ttest$var_respuesta, levels = niveles, ordered = TRUE),
              #factor(predictionsvmpol, levels = niveles, ordered = TRUE))
#acierto <- sum(diag(MC))/sum(MC)
#error <- 1 - acierto
#errorsvmpoli <- errorsvmpoli + error

#new_catsvmpol <- data.frame(id = TestId, predictionsvmpol)
#new_catsvmpol.ALL <- rbind(new_catsvmpol.ALL, new_catsvmpol)
## SVM Núcleo radial

modsvmrad <- svm(var_respuesta ~ ., data = train,
            kernel = "radial",
            type = "C-classification")


predictionsvmrad <- predict(modsvmrad, ttest)

MC <- table(factor(ttest$var_respuesta, levels = niveles, ordered = TRUE),
              factor(predictionsvmrad, levels = niveles, ordered = TRUE))
acierto <- sum(diag(MC))/sum(MC)
error <- 1 - acierto
errorsvmradi <- errorsvmradi + error

new_catsvmrad <- data.frame(id = TestId, predictionsvmrad)
new_catsvmrad.ALL <- rbind(new_catsvmrad.ALL, new_catsvmrad)

## SVM Núcleo sigmoideo

modsvmsig <- svm(var_respuesta ~ ., data = train,
```

```
            kernel = "sigmoid",
            type = "C-classification")


predictionsvmsig <- predict(modsvmsig, ttest)

MC <- table(factor(ttest$var_respuesta, levels = niveles, ordered = TRUE),
            factor(predictionsvmsig, levels = niveles, ordered = TRUE))
acierto <- sum(diag(MC))/sum(MC)
error <- 1 - acierto
errorsvmsigi <- errorsvmsigi + error

new_catsvmsig <- data.frame(id = TestId, predictionsvmsig)
new_catsvmsig.ALL <- rbind(new_catsvmsig.ALL, new_catsvmsig)

## Bosque aleatorio

modRF = randomForest(var_respuesta ~ .,
            data = train,
            ntree = as.numeric(Bosques.calibracion$bestTune[1]),
            mtry = dim(train)[2]*0.8,
            replace = TRUE,
            type = "classification")

predictionRF <- predict(modRF, ttest)

MC <- table(factor(ttest$var_respuesta, levels = niveles, ordered = TRUE),
            factor(predictionRF, levels = niveles, ordered = TRUE))
acierto <- sum(diag(MC))/sum(MC)
error <- 1 - acierto
errorRFi <- errorRFi + error

new_catrf <- data.frame(id = TestId, predictionRF)
new_catrf.ALL <- rbind(new_catrf.ALL, new_catrf)

## Clasificador ingenuo de Bayes

Bayes <- naiveBayes(var_respuesta ~ .,
             data = train.factor)

predictionBayes <- predict(Bayes, ttest.factor)

MC <- table(factor(ttest$var_respuesta, levels = niveles, ordered = TRUE),
            factor(predictionBayes, levels = niveles, ordered = TRUE))
acierto <- sum(diag(MC))/sum(MC)
```

```
error <- 1 - acierto
errorBayesi <- errorBayesi + error

new_catBayes <- data.frame(id = TestId, predictionBayes)
new_catBayes.ALL <- rbind(new_catBayes.ALL, new_catBayes)

## Vecinos más cercanos

knn <- train.kknn(var_respuesta ~ .,
                  data = train,
                  kmax = k.optimo)

predictionknn <- predict(knn, ttest)

MC <- table(factor(ttest$var_respuesta, levels = niveles, ordered = TRUE),
            factor(predictionknn, levels = niveles, ordered = TRUE))
acierto <- sum(diag(MC))/sum(MC)
error <- 1 - acierto
errorknni <- errorknni + error

new_catknn <- data.frame(id = TestId, predictionknn)
new_catknn.ALL <- rbind(new_catknn.ALL, new_catknn)

## XGBoost

## Número de clases para XGBoost
num.clases <- as.numeric(
      dplyr::count(distinct(train, var_respuesta)))


#train$var_respuesta.1 <- as.factor(as.numeric(as.character(train$var_respuesta)) - 1)
train$var_respuesta.1 <- as.factor(as.numeric(train$var_respuesta) - 1)
ttest$var_respuesta.1 <- as.factor(as.numeric((ttest$var_respuesta)) - 1)

sparse.data.train <- model.matrix(var_respuesta.1 ~ . -1,
                          data =  dplyr::select(train,-var_respuesta))

output_vector = matrix(train[,"var_respuesta.1"])


modelo.xgboost <- xgboost(params = parametrosXGBoost,
                    data = sparse.data.train,
                    label = output_vector,
                    verbose = 0,
                    nrounds = as.numeric(XGBoost.calibracion$bestTune[1]),
```

```
                        num_class = num.clases)

        sparse.data.test <- model.matrix(var_respuesta~ . -1,
                                data = dplyr::select(ttest,-var_respuesta))

        pred.xgboost <- predict(modelo.xgboost, sparse.data.test)

        pred.xgboost <- matrix(pred.xgboost, ncol=num.clases, byrow=TRUE)
        pred_xgboost <- max.col(pred.xgboost) - 1

        etiquetas <- unique(data.frame(original = train$var_respuesta, pred_xgboost =
train$var_respuesta1))
        etiquetas$pred_xgboost <- as.numeric(as.character(etiquetas$pred_xgboost))

        pred_xgboost.Data <- data.frame(pred_xgboost = pred_xgboost)
        XGBoost.Pred <- left_join(pred_xgboost.Data, etiquetas, by = "pred_xgboost")

        MC <- table(factor(ttest$var_respuesta, levels = niveles, ordered = TRUE),
                    factor(XGBoost.Pred$original, levels = niveles, ordered = TRUE))
        acierto <- sum(diag(MC))/sum(MC)
        error <- 1 - acierto
        errorxgboosti <- errorxgboosti + error

        new_catxgboost <- data.frame(id = TestId, XGBoost.Pred$original)
        new_catxgboost.ALL <- rbind(new_catxgboost.ALL, new_catxgboost)

        print(paste("Está en el fold", j), sep = " ")

    }
  #toc()

  errorsvmlini.CV[i] <- errorsvmlini / cv.folds
  #errorsvmpoli.CV[i] <- errorsvmpoli / cv.folds
  errorsvmradi.CV[i] <- errorsvmradi / cv.folds
  errorsvmsigi.CV[i] <- errorsvmsigi / cv.folds
  errorRFi.CV[i] <- errorRFi / cv.folds
  errorBayesi.CV[i] <- errorBayesi / cv.folds
  errorknni.CV[i] <- errorknni / cv.folds
  errorxgboosti.CV[i] <- errorxgboosti / cv.folds

  ResultadoVerdadero.svmlin.CV <- inner_join(ResultadoVerdadero.svmlin.CV,
new_catsvmlin.ALL, by = "id")
  #ResultadoVerdadero.svmpol.CV <- inner_join(ResultadoVerdadero.svmpol.CV,
new_catsvmpol.ALL, by = "id")
  ResultadoVerdadero.svmrad.CV <- inner_join(ResultadoVerdadero.svmrad.CV,
```

```
new_catsvmrad.ALL, by = "id")
  ResultadoVerdadero.svmsig.CV <- inner_join(ResultadoVerdadero.svmsig.CV,
new_catsvmsig.ALL, by = "id")
  ResultadoVerdadero.rf.CV <- inner_join(ResultadoVerdadero.rf.CV, new_catrf.ALL, by = "id")
  ResultadoVerdadero.Bayes.CV <- inner_join(ResultadoVerdadero.Bayes.CV, new_catBayes.ALL,
by = "id")
  ResultadoVerdadero.knn.CV <- inner_join(ResultadoVerdadero.knn.CV, new_catknn.ALL, by =
"id")
  ResultadoVerdadero.xgboost.CV <- inner_join(ResultadoVerdadero.xgboost.CV,
new_catxgboost.ALL, by = "id")

  }

  precision <- 1 - data.frame(errorsvmlini.CV, errorsvmradi.CV, errorsvmsigi.CV,
                    errorRFi.CV, errorBayesi.CV, errorknni.CV, errorxgboosti.CV)
  precision$iter.num <- seq(from = 1 , to = dim(precision)[1], by = 1)




  SVMLIN <- table(ResultadoVerdadero.svmlin.CV$resp_verdadera,
              factor(apply(convertidor.numerico(ResultadoVerdadero.svmlin.CV)[ ,
3:(cv.repetir+2)],
                    1, median), levels = niveles, ordered = TRUE))
  #SVMPOL <- table(ResultadoVerdadero.svmpol.CV$resp_verdadera,
   #              factor(apply(convertidor.numerico(ResultadoVerdadero.svmpol.CV)[ ,
3:(cv.repetir+2)],
    #                  1, median), levels = niveles, ordered = TRUE))
  SVMRAD <- table(ResultadoVerdadero.svmrad.CV$resp_verdadera,
              factor(apply(convertidor.numerico(ResultadoVerdadero.svmrad.CV)[ ,
3:(cv.repetir+2)],
                    1, median), levels = niveles, ordered = TRUE))
  SVMSIG <- table(ResultadoVerdadero.svmsig.CV$resp_verdadera,
              factor(apply(convertidor.numerico(ResultadoVerdadero.svmsig.CV)[ ,
3:(cv.repetir+2)],
                    1, median), levels = niveles, ordered = TRUE))
  RF <- table(ResultadoVerdadero.rf.CV$resp_verdadera,
            factor(apply(convertidor.numerico(ResultadoVerdadero.rf.CV)[ , 3:(cv.repetir+2)],
                    1, median), levels = niveles, ordered = TRUE))
  BAY <- table(ResultadoVerdadero.Bayes.CV$resp_verdadera,
             factor(apply(convertidor.numerico(ResultadoVerdadero.Bayes.CV)[ , 3:(cv.repetir+2)],
                    1, median), levels = niveles, ordered = TRUE))
  KNN <- table(ResultadoVerdadero.knn.CV$resp_verdadera,
             factor(apply(convertidor.numerico(ResultadoVerdadero.knn.CV)[ , 3:(cv.repetir+2)],
```

```
                    1, median), levels = niveles, ordered = TRUE))
 XGBoost <- table(ResultadoVerdadero.xgboost.CV$resp_verdadera,
              factor(apply(convertidor.numerico(ResultadoVerdadero.xgboost.CV)[ ,
3:(cv.repetir+2)],
                    1, median), levels = niveles, ordered = TRUE))


 return(
 DATA <-list(
          precision.total = apply(precision, 2, mean),
          precisiones = precision,
          svmlin.mc = SVMLIN,
          #svmpol.mc = SVMPOL,
          svmrad.mc = SVMRAD,
          svmsig.mc = SVMSIG,
          rf.mc = RF,
          bayes.mc = BAY,
          knn.mc = KNN,
          xgboost.mc = XGBoost,
          svmlin.pre.cat = medidas.precision(SVMLIN),
          #svmpol.pre.cat = medidas.precision(SVMPOL),
          svmrad.pre.cat = medidas.precision(SVMRAD),
          svmsig.pre.cat = medidas.precision(SVMSIG),
          rf.pre.cat = medidas.precision(RF),
          bayes.pre.cat = medidas.precision(BAY),
          knn.pre.cat = medidas.precision(KNN),
          xgboost.pre.cat = medidas.precision(XGBoost),
          plot.iter =
            ggplot(data = precision) +
                geom_line(aes(x= iter.num, y = errorsvmlini.CV, col = "SVM (Lineal)"), size = 1) +
                #geom_line(aes(x= iter.num, y = errorsvmpoli.CV, col = "SVM(Poli)"), size = 1) +
                geom_line(aes(x= iter.num, y = errorsvmradi.CV, col = "SVM (Radial)"), size = 1) +
                geom_line(aes(x= iter.num, y = errorsvmsigi.CV, col = "SVM (Sigmoideo)"), size =
1) +
                geom_line(aes(x= iter.num, y = errorRFi.CV, col = "Bosques Aleatorios"), size = 1)
+
                geom_line(aes(x= iter.num, y = errorBayesi.CV, col = "Bayes"), size = 1) +
                geom_line(aes(x= iter.num, y = errorknni.CV, col = "KNN"), size = 1) +
                geom_line(aes(x= iter.num, y = errorxgboosti.CV, col = "XGBoost"), size = 1) +
                labs(x = "Iteracción",
                    y = "Precisión",
                    col = "Modelos") +
                theme_bw()

          )
```

```
    )
  }
```

Función para medidas de precisión

```
medidas.precision <- function(MC)
  {

  precision.global <- round((sum(diag(MC))/sum(MC))*100, 2)
  precision.categoria <- round((diag(MC)/colSums(MC))*100, 2)
  conteo_predicho <- colSums(MC)

  resultado <- data.frame(etiqueta = names(precision.categoria),
                          precision.categoria,
                          conteo_predicho)
  return(resultado)

}
```

Función para rendimiento de modelo

```
resumen.modelos <- function(datos, pregunta_cod, objeto_predicciones)
  {

  datos$cod_resp <- as.numeric(as.character(eval(substitute(pregunta_cod), datos)))

  P <- datos %>%
          group_by(etiqueta) %>%
          summarise(conteo_original = n(), etiqueta_original = mean(cod_resp)) %>%
          arrange(desc(conteo_original))

  P$etiqueta <- str_to_title(P$etiqueta)

   Y <- objeto_predicciones
   colnames(Y)[1] <- "etiqueta_original"
   Y$etiqueta_original <- as.numeric(as.character(Y$etiqueta_original))
   Z <- left_join(P, Y, by = "etiqueta_original")
   Z$conteo_predicho <- coalesce(Z$conteo_predicho)

   original <- data.frame(Categoria = Z$etiqueta, Conteo = Z$conteo_original)
   original <- arrange(original, desc(Conteo))
   original$dist_pct <- round(original$Conteo / sum(original$Conteo) * 100, 2)
   original$Conteo <- NULL
   predicho <- data.frame(Categoria = Z$etiqueta, Conteo = Z$conteo_predicho,
Z$precision.categoria)
   predicho$Conteo <- coalesce(round(predicho$Conteo / sum(predicho$Conteo) * 100, 2),0)
```

```
   predicho$Z.precision.categoria <- coalesce(predicho$Z.precision.categoria, 0)
   predicho <- arrange(predicho, desc(Conteo))

  Data <- data.frame(original, predicho)
  colnames(Data) <- c("Categoría", "Dist Porcentual (%)", "Categoría.2", "Dist Porcentual (%).2",
"Precisión (%)")

  return(
          Data %>%
             mutate(`Dist Porcentual (%)` = color_bar("#a3f3ff")(`Dist Porcentual (%)`),
                  `Dist Porcentual (%).2` = color_bar("#a3f3ff")(`Dist Porcentual (%).2`),
                  `Precisión (%)` = color_bar("#da5b5b61")(`Precisión (%)`)) %>%
         select(Categoría, `Dist Porcentual (%)`, Categoría.2, `Dist Porcentual (%).2`, `Precisión
(%)`) %>%
          kable(escape = F, align = c("l", "r", "l", "r","r")) %>%
           kable_styling("hover", bootstrap_options ="striped") %>%
            add_header_above(c("Real" =  2, "Predicho" = 3)) %>%
            add_header_above(c("Comparación de clases reales con predichas" = 5))
      )
 }
```

Función de red de bigrama interactivo

```
red.bigramas.Vis <- function(Datos.Verificacion,
                      frecuencia.bigramas)
  {

   texto <- data.frame(respuesta = unlist(Datos.Verificacion$final_limpio))

  bigramas <- texto %>%
              unnest_tokens(bigram, respuesta,
              token = "ngrams",
              n = 2)

  bigramas_separados <- bigramas %>%
                      separate(bigram, c("word1", "word2"),
                            sep = " ")

  edges <- bigramas_separados %>%
                    count(word1, word2,
                       sort = TRUE) %>%
                       filter(n > frecuencia.bigramas)

   nodes <- data.frame(unique(c(edges$word1, edges$word2)))
   colnames(nodes) <- c("id")
```

```
  colnames(edges) <- c("from", "to", "value")
  nodes$label <- nodes$id

  nodes$font.color <- "white"

  return(
  visNetwork(nodes, edges,
          background = "black") %>%
   visEdges(arrows = 'to',
         color = list(color = "lightblue",
                      highlight = "red"))
       )
}
```

Función para red de correlaciones

```
red.correlaciones <- function(Datos.Verificacion, correlacion)
  {

  texto <- data.frame(respuesta = unlist(Datos.Verificacion$final_limpio))

  unnest_respuestas <- texto %>%
                    mutate(section = row_number()) %>%
                    unnest_tokens(word, respuesta,
                                  token = "ngrams",
                                  n = 1)

  word_cors <- unnest_respuestas %>%
               group_by(word) %>%
               filter(n() >= 5) %>%
               pairwise_cor(word, section,
                            sort = TRUE)
  return(
  word_cors %>%
   filter(correlation > correlacion) %>%
   graph_from_data_frame() %>%
   ggraph(layout = "fr") +
   geom_edge_link(aes(#edge_alpha = correlation,
                      edge_width = correlation,
                      colour = correlation)) +
   geom_node_point(color = "#29e05d7a", size = 5) +
   geom_node_text(aes(label = name), repel = TRUE) +
   theme_void()
      )

  }
```

```
### Función para dendrograma rectangular
```

```
arbol.dendrograma.normal <- function(cuerpo,
                              frecuencia.palabras, numero.grupos)
 {

  datos <- MatrizPalabraDocumento(cuerpo)

    num.col <- NULL
     for (k in 1:dim(datos)[1])
       {

         suma <- sum(datos[k , ])

           if(suma <= frecuencia.palabras)
             {

             num.col[k] <- k
             }
       }

   datos <- datos[-as.vector(na.omit(num.col)) , ]

  hc_euclidea <- hclust(d = dist(x = datos,
                                 method = "euclidean"),
                        method = "ward.D2")

  return(
  fviz_dend(x = hc_euclidea, k = numero.grupos,
          cex = 0.6,
          rect = TRUE,
          k_colors = "jco",
          rect_border = "jco", rect_fill = TRUE) +
    labs(title = "Clusters Jerárquico",
        subtitle = paste("Distancia euclídea, K =",
                         numero.grupos, sep = " "))
       )

 }
```

```
### Función para dendrograma filogenético
```

```
arbol.dendrograma <- function(cuerpo, frecuencia.palabras, numero.grupos)
 {

  datos <- MatrizPalabraDocumento(cuerpo)
```

```
  num.col <- NULL
   for (k in 1:dim(datos)[1])
     {

      suma <- sum(datos[k , ])

        if(suma <= frecuencia.palabras)
          {

          num.col[k] <- k
          }
     }

  datos <- datos[-as.vector(na.omit(num.col)) , ]

 hc_euclidea <- hclust(d = dist(x = datos,
                                method = "euclidean"),
                       method = "ward.D2")

 return(
 fviz_dend(x = hc_euclidea, k = numero.grupos, cex = 0.8,
         k_colors = "jco",
         type = "phylogenic", repel = TRUE) +
  labs(title = "Clusters Jerárquico - Árbol filogenético",
       subtitle = paste("Distancia euclídea, K =", numero.grupos, sep = " "))
      )

}
```

Función para red de bigramas

```
red.bigramas <- function(Datos.Verificacion, frecuencia.bigramas)
  {

 texto <- data.frame(respuesta = unlist(Datos.Verificacion$final_limpio))

 bigramas <- texto %>%
             unnest_tokens(bigram, respuesta,
             token = "ngrams",
             n = 2)

 bigramas_separados <- bigramas %>%
                       separate(bigram, c("word1", "word2"),
                             sep = " ")
```

```
bigrama_conteo <- bigramas_separados %>%
                  count(word1, word2,
                      sort = TRUE)


bigrama_grafico <- bigrama_conteo %>%
                  filter(n > frecuencia.bigramas) %>%
                  igraph::graph_from_data_frame()


formato <- grid::arrow(type = "closed", length = unit(.15, "inches"))

 Grado.Total = degree(bigrama_grafico, mode = "Total")

return(
ggraph::ggraph(bigrama_grafico, layout = "fr") +
  geom_edge_link(aes(edge_alpha = n), show.legend = FALSE,
            arrow = formato, end_cap = circle(.07, 'inches')) +
  #geom_node_point(color = "lightblue", size = 5) +
  geom_node_point(aes(size = Grado.Total, colour = Grado.Total)) +
  geom_node_text(aes(label = name), vjust = 1, hjust = 1) +
  theme_void()
     )

}
```

Indicadores de nodo para la red de bigramas

```
indicadores.nodo.red <- function(Datos.Verificacion, num.palabras)
  {

  texto <- data.frame(respuesta = unlist(Datos.Verificacion$final_limpio))

  bigramas <- texto %>%
              unnest_tokens(bigram, respuesta,
              token = "ngrams",
              n = 2)

  bigramas_separados <- bigramas %>%
                        separate(bigram, c("word1", "word2"),
                              sep = " ")


  bigrama_conteo <- bigramas_separados %>%
                    count(word1, word2,
```

```
                    sort = TRUE)


 bigrama_grafico <- bigrama_conteo %>%
                    filter(n > 3) %>%
                    igraph::graph_from_data_frame()

#Indicadores de nodo

 bonacich.eigen <- evcent(bigrama_grafico)$vector
 intermed <- betweenness(bigrama_grafico)
 #proximidad <- closeness(bigrama_grafico)
 indegree <- degree(bigrama_grafico,mode="in")
 outdegree <- degree(bigrama_grafico,mode="out")
 totaldegree <- degree(bigrama_grafico,mode="total")

 resumen <- cbind(totaldegree,indegree, outdegree)
 resumen <- resumen[order(resumen[,1], decreasing = TRUE),]

 rownames(resumen) <- str_to_title(row.names(resumen))

 colnames(resumen) <- c("Grado total", "Grado interno", "Grado externo")

 return(
 kable(resumen[1:num.palabras, ], "html") %>%
  kable_styling(bootstrap_options = c("striped", "hover")) %>%
   add_header_above(c("Indicadores de grado del nodo para la red de bigramas" = 4))
     )
 }
```

```
### Frecuencia palabras unigrama
```

```
frequencia.palabras.unigrama <- function(cuerpo,
                                maxima_frecuencia,
                                titulo)
 {

 tdm <- as.matrix(
           TermDocumentMatrix(cuerpo)
          )

 frequencia <- sort(rowSums(as.matrix(tdm)), decreasing = TRUE)
 datos <- data.frame(palabras = names(frequencia), conteo = frequencia)
 datos$palabras <- str_to_title(datos$palabras)

 subset(datos, conteo > maxima_frecuencia)    %>%
```

```
        ggplot(aes(reorder(palabras, conteo), conteo)) +
        geom_bar(stat = "identity", fill = "#87CEFA", colour = "#20B2AA") +
        geom_text(aes(label = conteo), hjust = -0.5) +
        coord_flip(ylim = c(0, max(datos$conteo)+6)) +
        theme_bw() +
        labs(y = "Frecuencia",
          x = " ",
         title = titulo,
         caption = "Fuente: Elaborado con datos de la ENPT 2019. CGR") +
        theme(plot.caption = element_text(hjust = 0))
}
```

Frecuencia de bigramas

```
frequencia.palabras.bigrama <- function(cuerpo,
                                maxima_frecuencia,
                                titulo)
{

 BigramTokenizer <- function(x) NGramTokenizer(x, Weka_control(min = 2, max = 2))

 tdm <- as.matrix(
            TermDocumentMatrix(cuerpo,
                    control = list(tokenize = BigramTokenizer))
           )

 frequencia <- sort(rowSums(as.matrix(tdm)), decreasing = TRUE)
 datos <- data.frame(palabras = names(frequencia), conteo = frequencia)
 datos$palabras <- str_to_title(datos$palabras)

 subset(datos, conteo > maxima_frecuencia)    %>%
        ggplot(aes(reorder(palabras, conteo), conteo)) +
        geom_bar(stat = "identity", fill = "#87CEFA", colour = "#20B2AA") +
        geom_text(aes(label = conteo), hjust = -0.5) +
        coord_flip(ylim = c(0, max(datos$conteo)+2)) +
        theme_bw() +
        labs(y = "Frecuencia",
          x = " ",
         title = titulo,
         caption = "Fuente: Elaborado con datos de la ENPT 2019. CGR") +
        theme(plot.caption = element_text(hjust = 0))
 }
```

Función para conteo de palabras en sentimientos

```
conteo.palabras.sentimientos <- function(Data)
  {

  Sent.Puntajes.C1.10 <- filter(Data, puntajes > 0)
  texto <- data.frame(respuesta = unlist(Sent.Puntajes.C1.10$texto))

  texto$respuesta <- as.character(texto$respuesta)
  colnames(lexicon.S1)[1] <- "word"

  texto <- texto %>%
        mutate(section = row_number()) %>%
        ungroup() %>%
        unnest_tokens(word, respuesta) %>%
        inner_join(lexicon.S1, by = "word") %>%
        count(word, sort = TRUE) %>%
        filter(n > 15)

  return(suppressWarnings(
  ggplot(texto, aes(reorder(word,n), n)) +
        geom_bar(stat = "identity", fill = "#4caf50b0") +
        geom_text(aes(label = n), hjust = -0.4) +
        coord_flip(ylim = c(1,max(texto$n)+11)  ) +
        theme_bw() +
        labs(y = "Puntaje",
          x = " ",
          title = "Frecuencia de palabras negativas",
          caption = "Fuente: Elaborado con datos de la ENPT 2019. CGR") +
    theme(plot.caption = element_text(hjust = 0))
      ))
  }
```

Función para gráfico de etiquetas

```
grafico.cat <- function(datos, titulo)
  {

   P <- datos %>%
            group_by(etiqueta) %>%
            summarise(conteo = n()) %>%
            arrange(desc(conteo))

   P$etiqueta <- str_to_title(P$etiqueta)

#ggplotly(
   ggplot(P, aes(x = reorder(etiqueta, conteo), y = conteo)) +
```

```
    geom_bar(fill = "#FF6347", stat = "identity", alpha = 0.8) +
    geom_text(aes(label = conteo), hjust = -0.5) +
    theme_bw() +
    coord_flip(ylim = c(0, max(P$conteo)+6) ) +
    labs(y = "Frecuencia",
         x = "",
        title = titulo,
        #paste(titulo, ", (n = ", dim(datos)[1], ")", sep = ""),
         caption = "Fuente: Elaborado con datos de la ENPT 2019. CGR") +
  theme(plot.caption = element_text(hjust = 0)) #  )
}
```

Función para cambiar palabras

```
cambiar.palabras <- function(datos)
  {

  datos$nuevo_stem <- str_replace_all(datos$nuevo_stem, "NA", "internet")
  datos$nuevo_stem <- str_replace_all(datos$nuevo_stem, " esa", " interesa")
  datos$nuevo_stem <- str_replace_all(datos$nuevo_stem, " testan", " contestan")
  datos$nuevo_stem <- str_replace_all(datos$nuevo_stem, " filar", " filas")
  datos$nuevo_stem <- str_replace_all(datos$nuevo_stem, " indo", " información")
  datos$nuevo_stem <- str_replace_all(datos$nuevo_stem, " personar", " personal")
  datos$nuevo_stem <- str_replace_all(datos$nuevo_stem, " digital izar", " digital izar")
 return(datos)

  }
```

Función para cambiar palabras (después de limpiar el texto)

```
cambiar.terminos <- function(cuerpo)
  {

  cuerpo <- tm_map(cuerpo, content_transformer(function(x)
                    gsub(x, pattern = "tramit",
                        replacement = "trámite")))

  cuerpo <- tm_map(cuerpo, content_transformer(function(x)
                    gsub(x, pattern = "tramitar",
                        replacement = "trámite")))

  cuerpo <- tm_map(cuerpo, content_transformer(function(x)
                    gsub(x, pattern = "burocrático",
                        replacement = "burocracia")))

  cuerpo <- tm_map(cuerpo, content_transformer(function(x)
```

```
                    gsub(x, pattern = "acces",
                        replacement = "acceso")))

cuerpo <- tm_map(cuerpo, content_transformer(function(x)
                    gsub(x, pattern = "accesoo",
                        replacement = "acceso")))

cuerpo <- tm_map(cuerpo, content_transformer(function(x)
                    gsub(x, pattern = "especifico",
                        replacement = "específico")))

cuerpo <- tm_map(cuerpo, content_transformer(function(x)
                    gsub(x, pattern = "actualic",
                        replacement = "actualic")))

cuerpo <- tm_map(cuerpo, content_transformer(function(x)
                    gsub(x, pattern = "mas",
                        replacement = "más")))

cuerpo <- tm_map(cuerpo, content_transformer(function(x)
                    gsub(x, pattern = "digit izar",
                        replacement = "digitalizar")))


cuerpo <- tm_map(cuerpo, content_transformer(function(x)
                    gsub(x, pattern = "d social",
                        replacement = "redes sociales")))

cuerpo <- tm_map(cuerpo, content_transformer(function(x)
                    gsub(x, pattern = "acced",
                        replacement = "acceso")))

cuerpo <- tm_map(cuerpo, content_transformer(function(x)
                    gsub(x, pattern = "publico",
                        replacement = "público")))

cuerpo <- tm_map(cuerpo, content_transformer(function(x)
                    gsub(x, pattern = "in eficiencia",
                        replacement = "ineficiencia")))

cuerpo <- tm_map(cuerpo, content_transformer(function(x)
                    gsub(x, pattern = "intelectualizar",
                        replacement = "desactualizados")))

cuerpo <- tm_map(cuerpo, content_transformer(function(x)
```

```
                    gsub(x, pattern = "espiritualizar",
                        replacement = "desactualización")))

cuerpo <- tm_map(cuerpo, content_transformer(function(x)
                    gsub(x, pattern = "sugerent",
                        replacement = "sugerencias")))

cuerpo <- tm_map(cuerpo, content_transformer(function(x)
                    gsub(x, pattern = "des actualizada",
                        replacement = "desactualizada")))

cuerpo <- tm_map(cuerpo, content_transformer(function(x)
                    gsub(x, pattern = "intelectualizada",
                        replacement = "desactualizada")))

cuerpo <- tm_map(cuerpo, content_transformer(function(x)
                    gsub(x, pattern = "intelectualizado",
                        replacement = "desactualizada")))

cuerpo <- tm_map(cuerpo, content_transformer(function(x)
                    gsub(x, pattern = "tramitología",
                        replacement = "trámite")))

cuerpo <- tm_map(cuerpo, content_transformer(function(x)
                    gsub(x, pattern = "mitología",
                        replacement = "trámite")))

return(cuerpo)

}
```

```
### Otras
```

```
eliminando.palabras.training <- function(datos)
 {

  num.col <- NULL
    for (k in 1:dim(datos)[2])
      {

        sum <- sum(datos[ , k])

          if(sum == 0)
            {
```

```
            num.col[k] <- k
            }
      }

 datos[ , as.vector(na.omit(num.col))] <- NULL

   return(datos)

}
```