UNIVERSIDAD INTERNACIONAL DE LA RIOJA

Universidad Internacional de La Rioja (UNIR)

Escuela de Ingeniería

Máster en Análisis y Visualización de Datos Masivos

Score de gestión de llamadas en recuperación de cartera

Trabajo Fin de Máster

Presentado por: Loja Cajas, Jorge

Director: Mantecón García, José

Repositorio: https://github.com/jorgelojam/scorecallcenter

Ciudad: Cuenca

Fecha: 18/09/2017

Resumen

En el proceso de recuperación de cartera existe varios factores y canales de comunicación que intervienen en la gestión de recuperación. Este trabajo se centra en la valoración de la comunicación que se establece entre un gestor y el socio a través de un callcenter, y está orientado a calificar la respuesta del socio, enmarcándola en la aceptación o el rechazo ante la notificación realizada por el gestor indicándole que tiene valores pendientes de pago. Con estos resultados, se trata de ayudar a mejorar los protocolos de comunicación, además de brindar un historial de calificación para que en futuras llamadas cualquier gestor identifique la clase de socio con el que va a comunicarse.

Utilizando reconocimiento de voz y el análisis de sentimientos se determina la polaridad de la respuesta, a través de aprendizaje automático supervisado con algoritmos de clasificación obteniendo clasificadores buenos con una eficiencia superior al 80% para determinar la relevancia de la respuesta y posteriormente la polaridad de la misma.

Palabras Clave: reconocimiento de voz, análisis de sentimientos, clasificación binaria, minería de opinión

Abstract

There are many factors and communication channels that work together in the process of portfolio recovery. This project is focused on the communication assessment that is carried out between a manager and a partner through a call center. It seeks to rate the quality of the partner's answer based on the acceptance or rejection after the manager informs him about the unpaid values. With these results, the goal is to improve the communication protocols offering a rating history that can be used in future calls, so any manager will be aware and identify the type of partner with whom is dealing with.

Using speech recognition and feelings analysis is possible to determine the polarity of the answer through automatic supervised learning with rating algorithms obtaining good classifiers with an efficiency superior to an 80% to determine the relevance and polarity of the answer.

Keywords: speech recognition, sentiment analysis, binary classification, opinion mining

Índice de contenido

1. Introducción....1

2. Estado del Arte....6

3. Objetivos....12

3.1. Objetivo general....12

3.2. Objetivos específicos....12

3.3. Metodología del trabajo....13

3.3.1. Sistema de evaluación y criterios de éxito....14

3.3.2. Clasificador de relevancia....15

3.3.3. Clasificador de Polaridad....16

4. Desarrollo de un sistema de clasificación de sentimientos....17

4.1 Enfoque....17

4.2. Compresión de los datos....17

4.2.1. Descarga de los audios....18

4.3. Preparación de los datos....21

4.3.1. Transcripción....22

4.3.2. Herramientas de reconocimiento de voz....22

4.3.3. Pre procesado y creación del corpus inicial....29

4.3.4. Proceso de anotado....30

4.4. Modelado....36

4.4.1 Clasificador de relevancia....41

4.4.2. Clasificador de polaridad....52

4.5. Evaluación de resultados....57

4.5.1. Clasificador de relevancia....57

4.5.2. Clasificador de polaridad 59

4.6. Despliegue 62

5. Conclusiones y trabajo futuro 64

5.1 Conclusiones 64

5.2 Líneas de trabajo futuro 66

6. Bibliografía 68

Índice de ilustraciones

Ilustración 1 Interfaz de usuario de callmanager Cisco.....19

Ilustración 2 Listado de llamadas realizadas por el gestor de cobranzas en el sistema de callmanager Cisco.....19

Ilustración 3 Archivos de audio descargados desde el sistema de callmanager 20

Ilustración 4 Formato de correspondencia de archivo de audio y registro de callmanager ...20

Ilustración 5 Arreglo de documentos con formato JSON.....35

Ilustración 6 Estructura Naive Bayes.....38

Ilustración 7 “Hiperplano de separación óptimo y su margen asociado (máximo)” (Suárez, 2014).....39

Ilustración 8 “Función de decisión no lineal en el espacio del conjunto de ejemplos original a un espacio transformado” (Suárez, 2014) 40

Ilustración 9 Pipeline de procesamiento 45

Ilustración 10 Comparación de vector de palabras con términos de frecuencia.....45

Ilustración 11 Comparación Unigramas vs Bigramas.....46

Ilustración 12 Comparación del uso o no de stop words.....46

Ilustración 13 Palabras comunes utilizadas con stop words 47

Ilustración 14: Palabras comúnmente utilizadas sin stop words 47

Ilustración 15 Comparación del uso o no de stemming 48

Ilustración 16 Guía de selección de algoritmos de machine learning (Scikit-learn, s.f.) 49

Ilustración 17 Comparación de resultados de los algoritmos de clasificación basado en AUC 51

Ilustración 18 Comparación de los tiempos tomados de los algoritmos de clasificación 51

Ilustración 19 Nube de palabras con polaridad positiva.....52

Ilustración 20 Nube de palabras de la polaridad negativa.....53

Ilustración 21 Vector de palabras vs TFM.....54

Ilustración 22 Unigramas vs Bigramas en clasificador de polaridad 54

Ilustración 23 Comparación de resultados de los algoritmos de clasificación de polaridad...56

Ilustración 24 Tiempo de convergencia de los clasificadores de polaridad 56

Ilustración 25 ROC AUC Clasificador de relevancia 59

Ilustración 26 ROC AUC Clasificador de polaridad.....61

Ilustración 27 Procesamiento de clasificadores.....62

Índice de tablas

Tabla 1 Matriz de confusión 14

Tabla 2 Fórmulas de las métricas utilizadas (Sokolova y Lapalme, 2009) 15

Tabla 3 Formato de archivo de correspondencia 21

Tabla 4 Comparación de motores de reconocimiento de voz open source 24

Tabla 5 Resultados medidos en WER (Gaida et al., 2014) 26

Tabla 6 Tamaño del corpus inicial 36

Tabla 7 Ejemplo de vector de palabras 42

Tabla 8 Ejemplo de vector de palabras con TF 43

Tabla 9 Parámetros óptimos para el clasificador de relevancia 57

Tabla 10 Matriz de confusión del clasificador de relevancia 58

Tabla 11 Métricas del clasificador de relevancia 58

Tabla 12 Parámetros óptimos para el clasificador de polaridad 60

Tabla 13 Matriz de confusión del clasificador de polaridad 60

Tabla 14 Métricas del clasificador de polaridad 61

1. Introducción

Las campañas de llamadas que los callcenters realizan han sido tradicionalmente manejadas con el objetivo de dar un mensaje para ofrecer un producto, para ofrecer un servicio, para realizar una encuesta, para dejar una notificación, etc. Este tipo de comunicación casi siempre es en dos vías, en donde el gestor de callcenter realiza una propuesta, oferta, pregunta que tiene que ser asimilada por la persona con la que se comunica, la cual a su vez entrega una respuesta que puede ser registrada por el gestor en un sistema de información para dicho propósito.

La comunicación, suele ser grabada para control de calidad, respaldo del contacto establecido, respeto o garantía hacia el usuario, exigencia de instituciones internas o externas que regulan este medio de comunicación y futuro contacto con el usuario o clientes. Con la finalidad de controlar que el usuario reciba un buen trato por parte del gestor, o por imperativo legal, en los protocolos de trato al usuario se suele advertir al usuario de que la llamada se está grabando como medida de control. Además, las llamadas suelen ser supervisadas, ya sea por medio de un controlador que escucha las conversaciones mientras éstas se están desarrollando, o se suele recurrir a las grabaciones de las conversaciones de manera aleatoria.

De este control se desprende una valoración que identifica si el gestor atendió bien o mal al usuario basado en el protocolo que debía seguir. Esta valoración está llena de subjetividad, ya que intervienen factores personales, sociales, económicos, laborales tanto del supervisor como del gestor para otorgar una calificación. Es por eso, que los callcenters han optado por modificar sus protocolos e incluir al final de la conversación una encuesta donde se le solicita al usuario calificar el trato o la atención que recibió del gestor.

En este escenario el problema para calificar la atención sigue latente, ya que confiar en el usuario sigue siendo subjetivo, debido a que éste podría indicar que no recibió un buen trato por parte del gestor, a pesar de que se haya seguido el protocolo de buen trato y comunicación establecida. El problema radica en que el usuario podría entregar una evaluación falsa o negativa dependiendo a distintos factores emocionales que le afecten en ese momento, como por ejemplo que estar ocupado, de mal humor, etc.

En el desarrollo de este trabajo se desea abordar una calificación de la respuesta que da el usuario en el escenario que se describe a continuación:

En las instituciones financieras se originan créditos a sus clientes o socios, que nacen desde un proceso de solicitud, pasando por un proceso de análisis de riesgo y factibilidad,

aprobación y finalmente con la entrega del dinero al solicitante. El beneficiario de este crédito se compromete a realizar pagos periódicos para cubrir la deuda que obtuvo. En determinadas ocasiones estos compromisos de pagos adquiridos no son cumplidos por diversos factores propios del cliente o socio.

Al no cumplir con el pago de la deuda adquirida, las instituciones financieras deben realizar una gestión de recuperación. Las instituciones financieras disponen de un área encargada de esta gestión: el área de cobranzas, en donde existe personal que realiza llamadas o visitas personalizadas al lugar de residencia o trabajo del socio. El proceso completo y los criterios para establecer las llamadas o las visitas no son parte del ámbito de este trabajo.

Al realizar una llamada, el gestor de cobranzas trata de acceder al cliente o socio titular de la deuda, con la finalidad de transmitirle el mensaje indicándole el valor pendiente de pago, el cual debe ser abonado a la brevedad posible, para evitar incrementar aún más la deuda. Este mensaje sigue un protocolo preestablecido para que el cliente o socio tenga el mejor trato posible.

El socio puede responder de varias maneras ante esta situación: en el mejor de los casos puede encarar de manera responsable estableciendo un acuerdo de pago, o al contrario, indicar de una manera despreocupada que no tiene la voluntad de realizar el pago por motivos o razones que no son aceptables.

Este proceso de calificación podría realizarse de manera manual por medio de controladores y supervisores que escuchar aleatoriamente las conversaciones. En este caso, sólo se cubriría una parte muy reducida de estas grabaciones, dejando muchas conversaciones de varios socios y clientes sin evaluar y calificar.

Para calificar a todas las grabaciones es necesario desarrollar un proceso que pueda de manera automática analizar las grabaciones disponibles, entregando una valoración de como respondió el cliente o socio ante la llamada de gestión de cobranza. Al disponer de esta calificación, se puede tener un feedback para que los gestores de turno identifiquen como se ha comportado ese socio previamente y sepan cómo atenderle de acuerdo con la forma en la que ha respondido anteriormente.

De esta forma, se ayuda a elaborar nuevos protocolos y estrategias de comunicación, basándose en los tipos de respuesta, o a adaptar los protocolos ya establecidos para ajustarse al perfil de cada cliente o socio.

Para el procesamiento automático que se quiere realizar, se presentan algunos retos que tienen que ser evaluados y abordados para llegar al objetivo principal de este trabajo: Calificar la respuesta de un cliente o socio.

La manera en la que tradicionalmente nos hemos relacionado e interactuado con los ordenadores es por medio de instrucciones expresadas en formatos de texto. La información de la respuesta está contenida en un archivo de audio, por tanto, es necesario extraer esta respuesta para ser representada en un formato que sea entendido por un sistema de procesamiento automático que se ejecuta en un ordenador. Estos textos deben ser transformados, por medio de la extracción de sus características, a representaciones que se puedan cuantificar para elaborar un modelo de calificación de acuerdo a sus características más representativas.

En el campo de la inteligencia artificial se han desarrollado desde hace varios años atrás sistemas que permiten realizar el reconocimiento de voz, conocidos como Automatic Speech Recognition (ASR), que por medio del desarrollo de metodologías y técnicas permiten el reconocimiento y transcripción de un lenguaje hablado a un texto. En la aplicación práctica para el trabajo de fin de máster, se utiliza un sistema ASR para identificar con la mayor precisión posible las palabras que una persona dijo en la conversación con el gestor y los convierte en un texto escrito, con un grado de error que varía dependiendo de distintos factores como ruido, volumen, tono de voz, acento, entre otros.

Posterior a la transcripción a texto de las conversaciones telefónicas, se debe calificar este texto, con la finalidad de determinar la polaridad de la respuesta. Para ello se realiza un análisis de sentimiento, orientado a determinar el estado de ánimo de la persona que recibió la llamada por parte del gestor de cobranza.

Estas respuestas habitualmente se componen de contenido subjetivo y están asociadas a sentimientos, que pueden ser positivos, negativos o neutros, en función de la experiencia y conducta de la persona y del trato que recibió por parte del gestor.

Hoy en día las opiniones o comentarios en Internet influyen en el comportamiento que tiene un usuario frente a un producto, una marca o un servicio, lo que influye directamente en la imagen de las compañías. Un ejemplo práctico de esto, serían los comentarios que realizan los usuarios en sitios de comercio electrónico. Estas opiniones pueden ser utilizadas por otros usuarios para ayudar en la toma de decisiones al realizar una compra; o desde la perspectiva del comercio, para identificar los comentarios negativos con la finalidad de realizar procesos de adecuación y mejora.

En el ámbito empresarial, monitorizar y analizar las opciones, comentarios o respuestas que se tienen de los usuarios o clientes pasa a ser algo esencial en un entorno muy competitivo. Las empresas se enfrentan con mayor o menor dificultad al problema de analizar un gran volumen de opciones, comentarios o respuestas relativas a productos y servicios que pueden proceder de varios orígenes de datos con gran diversidad de formatos. Estos datos podrían proceder de sistemas de callcenter, Interactive Voice Response (IVR), redes sociales, etc. El gran volumen de datos descarta la posibilidad de realizar un análisis manual, llegando a la necesidad de realizar procesos automáticos que se encarguen de este análisis.

Con la ayuda del procesamiento de lenguaje natural (PLN), el área de la inteligencia artificial se convierte en el factor clave para extraer masivamente el significado residente en el lenguaje natural o humano de un comentario, opinión o respuesta.

Por medio de este trabajo, se propone establecer una aproximación a un modelo de calificación de las respuestas del socio o cliente ante la llamada de un gestor de cobranzas, estableciendo una secuencia de procesos que conectan la transcripción de un audio a texto, y la posterior valoración y clasificación de la polaridad que contiene el texto por medio de las técnicas de análisis de sentimientos. Al aplicar estas técnicas, los textos con opiniones, sentimientos y subjetividad dejan de ser datos no estructurados para convertirse en datos estructurados, con información para su análisis y clasificación, permitiendo formarlos parte de cualquier análisis cuantitativo de cualquier empresa.

Los análisis y modelos desarrollados en el presente trabajo tienen como propósito escribir un programa de software, que permita automatizar la revisión de las respuestas que dan los socios o clientes que reciben una llamada por parte de un gestor de cobranzas. Y así, entregar al gestor, un nuevo parámetro que le sea de valor y le ayude a identificar con qué tipo de persona está tratando al momento de abordarle y desarrollar la conversación telefónica. De esta manera se podrá obtener un buen resultado con el cliente o socio e identificar cuando no se consigan resultados positivos, para que se puedan diseñar protocolos y atender al cliente o socio de una manera diferenciada.

Con la finalidad de realizar el proceso de transcripción se hace un breve análisis de los motores de reconocimiento de voz entre software APIs cerradas y abiertas. De acuerdo a la información que se puedo identificar en el primer acercamiento con los archivos de audio, se determinó necesario implementar un filtro previo ya que no todas las respuestas contienen información necesaria para establecer la valoración. Este filtro determina la relevancia de la respuesta definiendo si la respuesta tiene o no la información necesaria, por medio de un modelo de clasificación binaria.

Para determinar la valoración de la respuesta del socio se establece un criterio de polaridad, es decir, se indica si el socio respondió de manera positiva o negativa, sin otros posibles valores o niveles. Esto se con el objetivo de obtener una variable de salida dicotómica. Se analiza diferentes modelos de clasificación, empezando con los más utilizados para el tipo de clasificación, como lo son los algoritmos SVM y Naives Bayes. Basándose en las métricas establecidas, se evalúan varios casos de prueba de los modelos para determinar si alcanzan el nivel establecido de precisión o confianza que se desea.

2. Estado del Arte

El lenguaje natural que utilizamos los humanos para comunicarnos está incluido dentro de la categoría de datos no estructurados. El procesamiento de lenguaje natural es una subárea de la Inteligencia Artificial y la Lingüística, centrado en estudiar los problemas derivados de la generación y comprensión automática del lenguaje natural, convirtiendo los datos no estructurados en datos estructurados sobre los cuales se puede realizar análisis cuantitativos (Chowdhury, 2003).

En la aplicación práctica del trabajo se distinguen dos sub áreas del procesamiento de lenguaje natural: el reconocimiento automático del habla y la minería de textos.

En el reconocimiento automático del habla, conocido también como voz a texto, el objetivo es establecer un enlace en la comunicación persona - sistema informático, y procesar la señal de voz emitida por el ser humano y reconocer la información contenida en el audio para convertirlas en texto.

El reconocimiento de voz se ha venido desarrollando desde hace varias décadas atrás, empezando con el reconocimiento de palabras aisladas con un vocabulario muy reducido, basados en modelos acústicos y fonéticos, hasta evolucionar a vocabularios sin límites de palabras, pasando desde el uso de modelos estadísticos, estocásticos, semánticos hasta aplicar algoritmos de machine learning para mejorar el reconocimiento (Juang y Rabiner, 2005).

En las áreas de aplicación prácticas del reconocimiento de voz están las comunicaciones telefónicas, sistemas de asistencia guiada, atención al cliente, centros de información, callcenters, y otros que se detallan a continuación:

- Dictado automático: es en donde más se han utilizado los procesos desarrollos de reconocimiento de voz, para dictados de oficios, textos legales, diagnósticos y recetas médicas. Con la finalidad de mejorar la precisión del sistema se utilizan corpus especializados.

- Control por comandos: sistemas diseñados para dar órdenes a dispositivos y programas que se ejecutan sobre estos. Al trabajar con comandos, el reconocimiento se limita a un vocabulario muy reducido, por lo cual estos sistemas tienen más precisión.

- Sistemas Private Branch Exchange (PBX): permiten a los usuarios ejecutar comandos mediante el habla en lugar de pulsar teclas, por ejemplo: al dictar el número a marcar.

- Sistemas móviles, teléfonos o relojes inteligentes: se utiliza el habla como un método para introducir datos en diferentes aplicaciones de los mismos.

- Sistemas de apoyo a discapacitados: son de ayuda para personas con necesidades especiales que tiene dificultad para teclear de manera continua o dificultades al escuchar (Rabiner y Juang, 1993).

La minería de textos se basa en el procesamiento de lenguaje natural escrito, aplicando un análisis de texto con la finalidad de convertir datos no estructurados cargados de subjetividad a datos estructurados para su evaluación e interpretación en la salida.

Entre las aplicaciones típicas de la minería de textos tenemos:

- Clasificación de documentos (Text categorization / Document classification): se clasifica un texto particular o un documento en general a una o más categorías o niveles previamente establecidos. Entre las aplicaciones prácticas tenemos: el filtrado de spam, identificación de idioma, análisis de sentimientos, etc. (Sebastiani, 2002).

- Similitud semántica de textos (Semantic Text Similarity): se compara dos textos para buscar su grado de similitud semántica. La aplicación práctica por excelencia está en los buscadores de Internet, y de manera general la búsqueda y recuperación de información.

- Resumen de documentos (Document summarization): se extrae un conjunto de datos representativos de un determinado documento por medio de la identificación de las partes más informativas. Un ejemplo práctico se da en Internet, cuando un buscador por medio de sus robots revisa el contenido de miles de páginas con la finalidad de presentar primero las más relevantes dentro de un tema.

- Extracción de conceptos o entidades (Named entity recognition): se localiza y clasifica los elementos de un texto, organizando en categorías predefinidas los textos que representan nombres de personas, de instituciones, de empresas o compañías, de lugares, de cantidades, etc.

- Desambiguación del significado (Word Sense Disambiguation): se utiliza para darle sentido a una palabra específica de un texto según el contexto en el que aparece,

por ejemplo: banco, que puede entenderse como una institución financiera, como un asiento en el cual se pueden sentar personas, un depósito en el cual se conservan y almacenan plantas, etc. dando sentido a la palabra en el contexto en el que aparece.

- Análisis de sentimientos (Sentiment Analysis/Opinion mining): se determina la polaridad general que tiene un documento, un comentario o una frase. Se trata de detectar la actitud de la persona que lo escribió de acuerdo a las posibles emociones, juicios o evaluaciones contenidas en el texto. La clasificación más común es la binaria, es decir una opinión puede ser positiva o negativa (Pang y Lee, 2008).

En Internet debido a la creciente presencia de medios de comunicación social como blogs, redes sociales, comercios electrónicos, entre otros, la aplicación del análisis de sentimientos es sumamente relevante con la finalidad de ser competitivos en el mundo empresarial. El análisis de revisiones permite examinar todo tipo de expresiones que transmiten los clientes: sus opiniones de productos, posicionamiento, imagen, reputación, entre otros, que conlleven a la permanencia y la generación de nuevas oportunidades de negocio (Blair-Goldensohn et al., 2008).

Pensar que la tarea de clasificar los sentimientos de un texto en positivo o negativo es un proceso simple es algo completamente alejado de la realidad. Existe una complicación implícita, en el simple hecho de que diferentes personas, revisando un comentario, logren ponerse de acuerdo en la polaridad que tiene el mismo, y poder asignarle una clasificación definitiva. Esta dificultad se presenta cuando un mismo texto, puede ser interpretado de forma diferente en función de diversos factores socio culturales, regionales, idiomáticos, incluso personales. Es por ello, por lo que, para poder extraer información objetiva, se debe disponer de un conjunto suficientemente amplio de opiniones.

Debido a la subjetividad, la clasificación de sentimientos es completamente dependiente del dominio en el cual se produjeron las opiniones, comentarios o respuestas. Uno de los principales retos en este campo es conseguir un buen rendimiento al clasificar independiente del dominio (Liu, 2010).

El problema de identificar la polaridad de una opinión o un comentario en un campo abierto en el área del procesamiento de lenguaje natural, cuyo principal objetivo es determinar la polaridad de las opciones a nivel de documento, frase o término, clasificando binariamente en positivo o negativo. Para resolver estos problemas se han aplicado varios enfoques:

- Un enfoque semántico (Turney, 2002) propone y se apoya en el uso de diccionarios semánticos o lexicones de opiniones. A cada palabra que expresa una opinión se le asigna una orientación semántica.

- Otro enfoque propone resolverlo como un problema genérico de clasificación basado en el aprendizaje automático (Pang, Lee y Vaithyanatha, 2002).

- Con el enfoque supervisado se trabaja con un conjunto de datos (dataset) en los cuales se va anotando la polaridad del documento, comentario o texto. Esta anotación individual puede hacerse de forma automática o de forma manual por medio de anotadores humanos.

Una de las ventajas de la anotación automática de textos es que, al disponer ya de una valoración, se puede conseguir y procesar un gran conjunto de datos para entregar al modelo, al contrario de la anotación manual, en donde el factor humano le agrega precisión al modelo, pero la cantidad de datos que se dispone para el entrenamiento es menor, y debe existir un pre procesamiento en el cual se requiere mayor intervención manual.

El problema de clasificación de textos puede ser resuelto con métodos de aprendizaje no supervisados. Bakliwal et al. (2012) presentó un método de evaluación de sentimientos no supervisado que alcanzó buenos resultados. Pero las técnicas de aprendizaje supervisado han sido las más usadas y documentadas. El análisis de sentimientos ha sido desarrollado principalmente por medio de un aprendizaje supervisado. De hecho, la clasificación de sentimientos es un problema de clasificación de textos, ya que existen palabras que determinan y entregan relevancia a la opinión, por ejemplo: excelente, bueno, malo, grandioso, entre otras (Bing Liu, 2012; Hatzivassiloglou y McKeown, 1997).

Los métodos de aprendizaje que pueden ser aplicados son: Naive Bayes y Support Vector Machine (SVM) (Joachims, 1998).

Pang Lee y Vaithyanathan (2002) son los autores del primer documento de investigación que aplicó un modelo de clasificación para separar las opiniones de un conjunto de datos de películas en dos clases: positiva y negativa. Se utilizó una representación de vectores de palabras (unigramas) como rasgos de clasificación, con los cuales se obtuvo buenos resultados con el Naive Bayes y SVM.

En investigaciones posteriores, se probaron otros algoritmos de aprendizaje, así como diferentes maneras de caracterizar la información, obteniendo nuevas variables independientes, en donde la clave para la clasificación de sentimientos es tener un conjunto de rasgos característicos efectivos. Algunas de estas características son:

- Términos y su frecuencia: palabras individuales (unigramas) y n-gramas con su frecuencia asociada. En algunos casos las posiciones de las palabras pueden ser

medidas en términos de su valor TF-IDF (Term frequency – Inverse document frequency) o calculada mediante técnicas estadísticas.

- Categorías gramaticales de las palabras: adjetivos, sustantivos, verbo, etc.
- Frases y palabras con sentimientos: palabras del lenguaje que expresan sentimientos positivos o negativos, por ejemplo: bueno, excelente, como palabras positivas; malo, feo, terrible como palabras negativas.
- Palabras modificadoras, inversoras de la polaridad, tales como: no, nunca, poco, nada, nadie, debería, podría, etc.

Turney (2002) realizó una de las primeras implementaciones de clasificación no supervisada, ejecutando una clasificación en algunos patrones sintácticos fijos con palabras consecutivas (bigramas) que se suelen utilizar para expresar opiniones. Para formarlos se toma como base la categoría gramatical de las palabras, a continuación, a estos bigramas se les asigna una orientación semántica o polaridad basada en la distancia a la palabra positiva “excelente” y a la distancia a la palabra negativa “pobre”. El cálculo de la polaridad del documento se calcula de la media de todas las polaridades conseguidas por los bigramas.

Otro enfoque no supervisado es el método basado en lexicones, mismo que usa un diccionario con frases y palabras sentimentales, las cuales tienen asociada una orientación, y una fuerza, además de una intensidad y una negación, que sirven para computar una medida de sentimiento de cada documento.

Tanto el enfoque no supervisado como el supervisado tienen sus beneficios y obstáculos ya que son líneas de investigación abiertas en el análisis de sentimientos.

Los problemas en el ámbito de investigación del análisis de sentimientos tienen principalmente tres niveles de profundidad o granularidad: documento, frase o sentencia y entidad y aspecto.

- Documento: el problema que se aborda en este nivel es el de como clasificar de manera general la opinión sobre un documento, si de manera positiva o negativa (Pang, Lee y Vaithyanathan, 2002; Turney, 2002). Por ejemplo, sobre la revisión de un producto, el sistema determina que las opiniones en conjunto expresan una opinión positiva o negativa del mismo; se asume que cada documento expresa opiniones sobre una única entrada.

- Frase o sentencia: en este nivel de análisis se considera cada frase o sentencia como una unidad única e independiente, expresada de manera positiva o negativa. Este nivel de análisis esta cercanamente relacionado a la clasificación de subjetividad, que busca determinar si una frase es subjetiva u objetiva (Wiebe, Bruce y O'Hara, 1999).

- Entidad y Aspecto: es el nivel de análisis más fino en las líneas de investigación actuales. La finalidad es lograr extraer la mayor cantidad de información de las opiniones, en lugar de buscar construcciones del lenguaje (documentos, párrafos, sentencias, cláusulas o frases). A nivel de aspecto se busca directamente la opinión como tal; se centra en la premisa que una opinión está compuesta de un sentimiento positivo o negativo y un objetivo. El objetivo de este nivel de análisis es descubrir los sentimientos sobre las entidades y/o sus aspectos (Yi et al., 2003; Nasukawa y Yi, 2003; Hiroshi et al., 2004).

Basándose en los estudios y trabajos desarrollados en los campos del procesamiento de lenguaje natural existe métodos y herramientas formales, que nos permiten: evaluar y calificar las expresiones, sentimientos, emociones expresadas por un usuario por medio de diferentes medios de comunicación.

3. Objetivos

En esta sección se explica el alcance del trabajo de fin de master mediante la definición de los objetivos generales y específicos.

3.1. Objetivo general

Diseñar un sistema que califique automáticamente la polaridad de las repuestas de un socio ante una llamada de un gestor de cobranza por medio del callcenter.

3.2. Objetivos específicos

El dominio escogido para realizar este trabajo de fin de master se centra en el reconocimiento automático del habla, análisis de sentimientos sobre las respuestas que se obtiene de un cliente al recibir una llamada de un gestor de cobranzas. Este análisis entregará una calificación que determinará si el cliente respondió de manera positiva o negativa. Para alcanzar el objetivo general se plantean los siguientes objetivos:

1. Identificar los motores de reconocimiento de voz disponibles, tanto implementaciones comerciales como opensource.
2. Analizar el rendimiento y exactitud de los motores de reconocimiento de voz, con pruebas de concepto unitarias.
3. Desarrollar el proceso de transcripción de voz a texto con el menor grado de error.
4. Seleccionar y descargar la muestra representativa de la campaña de gestión de cobranza.
5. Escribir un programa de que realice el proceso de transcripción automático de las conversaciones grabadas.
6. Estructurar el corpus inicial de las transcripciones en texto, obtenidas de los archivos de audio.
7. Definir el proceso de anotado manual de las respuestas del socio en el corpus inicial.
    a. Identificar si la respuesta contiene la información necesaria para realizar un análisis posterior.
    b. Identificar la polaridad de la respuesta del socio para determinar si esta es afirmativa o negativa.

8. Identificar los modelos de clasificación que procesen información contenida en textos.
9. Evaluar los métodos o técnicas de caracterización de información contenida en textos.
10. Evaluar los modelos de clasificación utilizados para aprendizaje automático supervisado.
11. Identificar las métricas más representativas para medir la efectividad de los modelos de clasificación.
12. Seleccionar el modelo de clasificación con mejores prestaciones para determinar la relevancia y posterior polaridad de la respuesta.
13. Escribir un programa que realice la clasificación automática de la polaridad de la respuesta del socio.
14. Establecer la tasa de contacto directa al socio dueño de la deuda.

3.3. Metodología del trabajo

El trabajo se basa en la experimentación de herramientas que permitan desarrollar programas para valorar la respuesta de un socio, la misma que se encuentra contenida en un archivo de audio. Por lo tanto, el análisis se hace sobre las grabaciones que contiene un callcenter, en la campaña de cobranza.

El análisis de las respuestas de los socios se realiza aplicando técnicas propias de la minería de textos y minería de opinión, por lo cual se necesita extraer los textos de los audios.

Se necesita identificar y establecer un motor de reconocimiento de voz que entregue buenos resultados al transcribir grabaciones de audio, teniendo en cuenta particularidades que se dan en el idioma español, además de las variaciones derivadas del acento, tono de voz, región, entre otros, que agregan complejidad a los motores de reconocimiento.

Se debe establecer una muestra representativa de las grabaciones que estén disponibles en el callcenter, para descargar estos archivos, que estos archivos deben ser procesados por el motor de reconocimiento de voz. Al ser una cantidad considerable, se debe crear un programa que automatice la transcripción, reciba la ubicación del archivo del audio y transcriba la información del archivo, segmentando la conversación en un texto que contiene

lo que dijo el gestor de cobranzas, y otro texto que contiene la respuesta que dio el socio. Caso contrario debe existir un proceso manual y operativo en donde se escuche las grabaciones y se digite el contenido de la conversación.

Se elabora el corpus inicial, que contiene ya la información estructurada resultado de la transcripción. Este corpus debe pasar por un proceso de anotado, que es manual, y realizado con criterios claramente definidos y unificados, de acuerdo al ámbito y objeto del estudio, entregando al anotador humano un proceso que le indique como interpretar y clasificar una respuesta.

Se debe considerar que no todas las repuestas que dan los socios contienen información relevante para el análisis de la respuesta. Por citar un ejemplo cotidiano, al llamar a un número de teléfono si la persona no está disponible, una grabación se reproduce indicando que la persona no puede contestar la llamada y que si desea puede dejar un mensaje. Ésta no es una respuesta propia del socio, pero consta en las grabaciones de audio, por lo tanto, esta información no es relevante y debe ser filtrada de forma previa a establecer una polaridad en la respuesta del socio.

Con la información completa en el corpus inicial, se aplican ya los modelos propios de machine learning para el aprendizaje automático supervisado, que son de clasificación binaria que permiten determinar la polaridad de una respuesta, de acuerdo a las características que tiene el texto de la respuesta.

En la experimentación de los clasificadores, se realizan pruebas para el establecimiento de las características de los textos, que aporten mayor precisión al modelo, comparando los resultados de acuerdo a métricas propias de los clasificadores binarios.

3.3.1. Sistema de evaluación y criterios de éxito

La exactitud de un modelo de clasificación puede ser evaluado utilizando la matriz de confusión que tiene la forma de un arreglo de la siguiente manera (Sokolova y Lapalme, 2009):

Clase	Clasificado como positiva	Clasificado como negativa
Positiva	Verdadero positivo (tp)	Falso negativo (fn)
Negativa	Falso positivo (fp)	Verdadero negativo (tn)

Tabla 1 Matriz de confusión

Se calcula el número de instancias que fueron correctamente clasificadas (verdadero positivo), el número de instancias que fueron correctamente clasificadas que no pertenecen a la clase (verdadero negativo), las instancias que se asignaron incorrectamente a una clase (falso positivo) o que no fueron reconocidos como instancias de la clase (falsos negativos).

Con los valores de la matriz de confusión se calculan la mayoría de las métricas que se utilizan en la clasificación binaria. Para este trabajo, además se utilizará una curva ROC (Reception Operating Characteristic) que es la manera más utilizada para visualizar el rendimiento de un clasificador binario y el AUC (área bajo la curva) que es probablemente la mejor manera de resumir el rendimiento del clasificador en un solo número.

Medida	Fórmula	Evaluación
Sensibilidad	$\frac{tp}{tp+fn}$	Efectividad del clasificador para reconocer clases positivas
Especificidad	$\frac{tn}{tn+fp}$	Que tan efectivo el clasificador identifica clases negativas
AUC	$\frac{1}{2}(\frac{tp}{tp+fn}+\frac{tn}{tn+fp})$	La capacidad que tiene el clasificador para evitar falsa clasificación

Tabla 2 Fórmulas de las métricas utilizadas (Sokolova y Lapalme, 2009)

3.3.2. Clasificador de relevancia

Como se mencionó anteriormente, el primer paso en la clasificación es verificar que la respuesta del socio contenga suficiente información importante, filtrando información no relevante que podría agregar ruido, para así asegurar la calidad del análisis de sentimientos posterior.

Se espera identificar una buena cantidad de respuestas relevantes con respecto a las que no lo son, para que el siguiente clasificador tenga información válida para un análisis más profundo. La métrica que se utilizará para determinar el rendimiento del clasificador de relevancia es el AUC, de donde, si se logra alcanzar un AUC de 0,9 en el conjunto de datos de entrenamiento, se considera como un clasificador muy bueno.

3.3.3. Clasificador de Polaridad

El análisis de sentimientos se realizará exclusivamente sobre las opiniones que han sido previamente catalogadas como relevantes por el clasificador anterior.

En este proceso se va a detectar y establecer la polaridad de las respuestas de los socios con una clasificación positiva o negativa. Se desea que el sistema clasifique de mejor manera las respuestas positivas y negativas, ya que esto ayudará al gestor a abordar al cliente de forma adecuada en las siguientes llamadas dependiendo de cuál haya sido su clasificación de respuesta. Cuando se encuentre con socios que tiene una clasificación positiva se podrá establecer una conversación fluida, amable y serena; a diferencia de cuando se trate con un socio que tiene una mala actitud, en donde el gestor tendrá que abordarle de una manera respetuosa, pero con firmeza.

Se espera que el número de las respuestas positivas será superior al número de las negativas, ya que en muchas de las ocasiones la llamada se establece con referidos del deudor, en donde se limitan a escuchar al gestor y establecen como compromiso el hacerle llegar el mensaje al deudor. En este caso se utilizará como métrica de evaluación el AUC, y se considerará como un clasificador bueno si se obtiene un valor de 0,9 o mayor con los datos de prueba.

4. Desarrollo de un sistema de clasificación de sentimientos

4.1 Enfoque

Como origen de datos para la construcción de los clasificadores se utiliza las grabaciones que se tienen disponibles en el callcenter utilizado por los gestores de cobranzas. Se extrae una muestra representativa, que es un conjunto de archivos de audio, de los cuales se extrae la máxima información disponible por medio del reconocimiento automático de voz, con la finalidad de reducir el tiempo en la transcripción manual del audio a texto.

El propósito principal es establecer de una manera automática la polaridad de la respuesta del socio, sobre todo la positiva, para reforzar la relación y mantener al socio. Dicha respuesta puede ser corta, o una serie de respuestas a las preguntas que hace el gestor, pudiendo contener sentimientos positivos y/o negativos en cada respuesta. Para resumir en un solo resultado, el análisis de sentimiento se lo hará a nivel de toda la respuesta en el contexto de la gestión de cobranzas.

Para llegar a clasificar la polaridad es necesario realizar un tratamiento previo de las respuestas que logremos extraer de los audios de las conversaciones, ya que existirá variedad de respuestas y ruido (que serían las respuestas irrelevantes) que no proveen ninguna información relevante para el clasificador de polaridad, por lo que hay que identificarlos, filtrarlos y excluirlos en el análisis de la polaridad. Bing Liu, (2012) cita los tipos de spam y spamming en donde existe el comentario irrelevante, que son llamados así porque no contienen una opinión o respuesta, o es un texto que no tiene nada que ver con el contexto.

4.2. Compresión de los datos

En esta etapa se realiza la primera aproximación a los datos que se va a analizar, en donde tenemos las etapas de recolección, transcripción, exploración y la verificación de la calidad de los mismos.

Los callcenters guardan las conversaciones que establecen con los clientes, ya sea por cuestiones de calidad o por exigencia del mercado o instituciones de control. Para la realización del trabajo, se ha solicitado acceso al sistema gestor de grabaciones, por medio de las cuales se puede acceder a descargar las grabaciones en archivos de audio.

4.2.1. Descarga de los audios

Para empezar a explorar los datos, se toma una muestra con las siguientes premisas:

- En el callcenter se definen campañas que responden a distintos propósitos como: promoción de servicios, venta de tarjetas de crédito, actualización de información y la gestión de cobranza para recuperar cartera por vencer o vencida, siendo este último el enfoque de nuestro estudio.

- Se realiza un promedio de 100 llamadas al día entre varios gestores. Esta cantidad se incremente en los últimos días del mes, cuando la mayoría de las cuotas tienen que ser pagadas.

Al tener 22 días laborales, se puede llegar a tener unas 3000 llamadas mensuales, contando con el incremento que ocurre al fin de mes. Si se va a realizar una clasificación tomando dos clases (positivo, negativo) se puede estimar el tamaño de la muestra, para empezar con el proceso de exploración, de la siguiente manera:

Para un nivel de confianza del 95% y una variabilidad máxima (P=0.5), el tamaño de la muestra puede calcularse aplicando esta fórmula propuesta por Israel (1992):

$$n = \frac{N}{1 + N(e)^2}$$

Dónde: N serían 3000 llamadas e = 0,05 para un error del 5% ó 0,03 para un error del 3%.

Con esto tenemos:

$$n = \frac{3000}{1 + 3000(0{,}05)^2} = 352{,}94$$

353 llamadas para un error del 5% con un nivel de confianza del 95%

$$n = \frac{3000}{1 + 3000(0{,}03)^2} = 810{,}81$$

811 llamadas para un error del 3% con un nivel de confianza del 95%

Esta fórmula se propone como método simplificado para calcular el tamaño de la muestra para variables dicotómicas.

De esta manera se descargará del callcenter 353 llamadas que serán el conjunto de datos inicial. En este caso se consiguió acceso al callcenter de Cisco como se muestra a continuación:

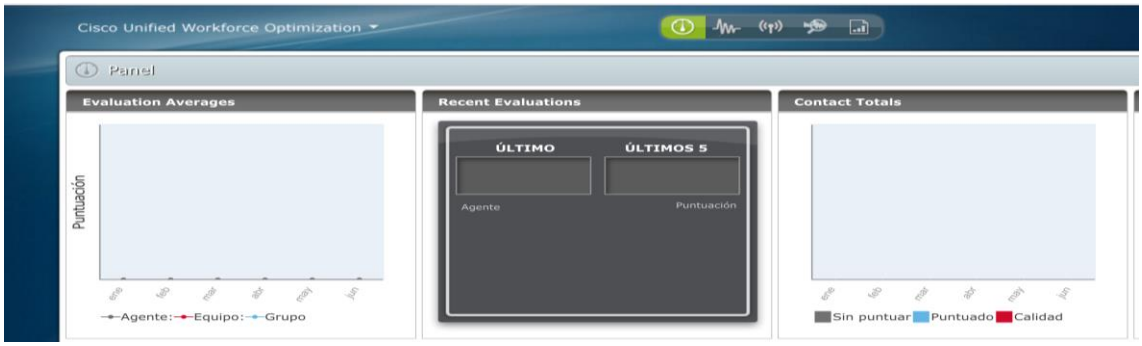
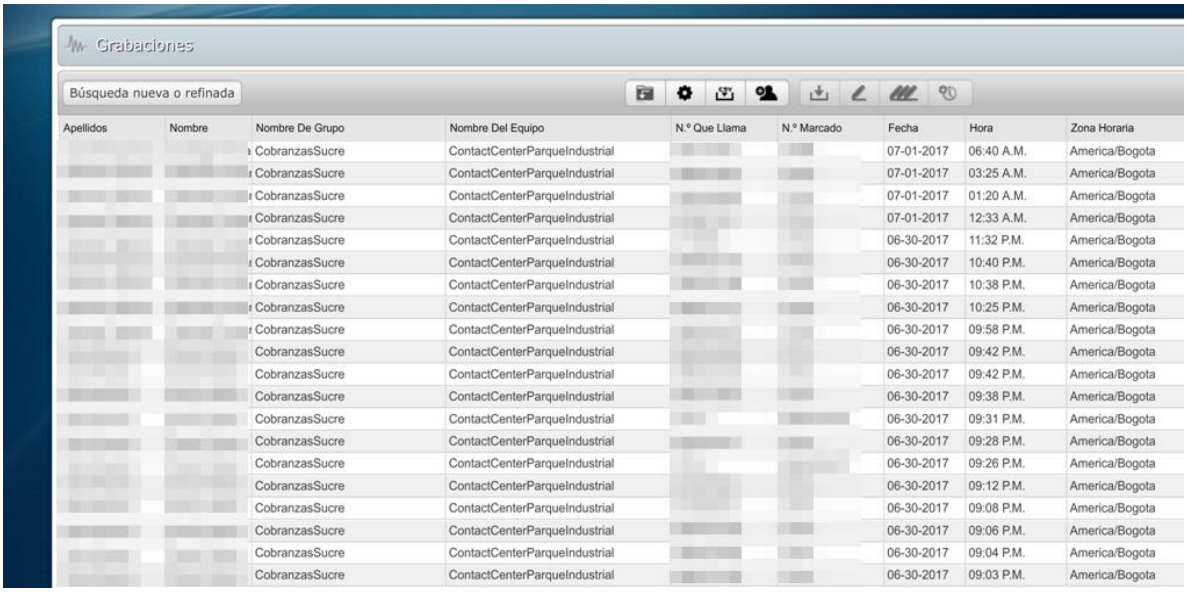


Ilustración 1 Interfaz de usuario de callmanager Cisco

Por medio de la interfaz web que dispone este callcenter (puede variar dependiendo del fabricante del software) se hace la descarga manual de la muestra previamente calculada desde una pantalla como la siguiente:



Apellidos	Nombre	Nombre De Grupo	Nombre Del Equipo	N.º Que Llama	N.º Marcado	Fecha	Hora	Zona Horaria
		CobranzasSucre	ContactCenterParqueIndustrial			07-01-2017	06:40 A.M.	America/Bogota
		CobranzasSucre	ContactCenterParqueIndustrial			07-01-2017	03:25 A.M.	America/Bogota
		CobranzasSucre	ContactCenterParqueIndustrial			07-01-2017	01:20 A.M.	America/Bogota
		CobranzasSucre	ContactCenterParqueIndustrial			07-01-2017	12:33 A.M.	America/Bogota
		CobranzasSucre	ContactCenterParqueIndustrial			06-30-2017	11:32 P.M.	America/Bogota
		CobranzasSucre	ContactCenterParqueIndustrial			06-30-2017	10:40 P.M.	America/Bogota
		CobranzasSucre	ContactCenterParqueIndustrial			06-30-2017	10:38 P.M.	America/Bogota
		CobranzasSucre	ContactCenterParqueIndustrial			06-30-2017	10:25 P.M.	America/Bogota
		CobranzasSucre	ContactCenterParqueIndustrial			06-30-2017	09:58 P.M.	America/Bogota
		CobranzasSucre	ContactCenterParqueIndustrial			06-30-2017	09:42 P.M.	America/Bogota
		CobranzasSucre	ContactCenterParqueIndustrial			06-30-2017	09:42 P.M.	America/Bogota
		CobranzasSucre	ContactCenterParqueIndustrial			06-30-2017	09:38 P.M.	America/Bogota
		CobranzasSucre	ContactCenterParqueIndustrial			06-30-2017	09:31 P.M.	America/Bogota
		CobranzasSucre	ContactCenterParqueIndustrial			06-30-2017	09:28 P.M.	America/Bogota
		CobranzasSucre	ContactCenterParqueIndustrial			06-30-2017	09:26 P.M.	America/Bogota
		CobranzasSucre	ContactCenterParqueIndustrial			06-30-2017	09:12 P.M.	America/Bogota
		CobranzasSucre	ContactCenterParqueIndustrial			06-30-2017	09:08 P.M.	America/Bogota
		CobranzasSucre	ContactCenterParqueIndustrial			06-30-2017	09:06 P.M.	America/Bogota
		CobranzasSucre	ContactCenterParqueIndustrial			06-30-2017	09:04 P.M.	America/Bogota
		CobranzasSucre	ContactCenterParqueIndustrial			06-30-2017	09:03 P.M.	America/Bogota

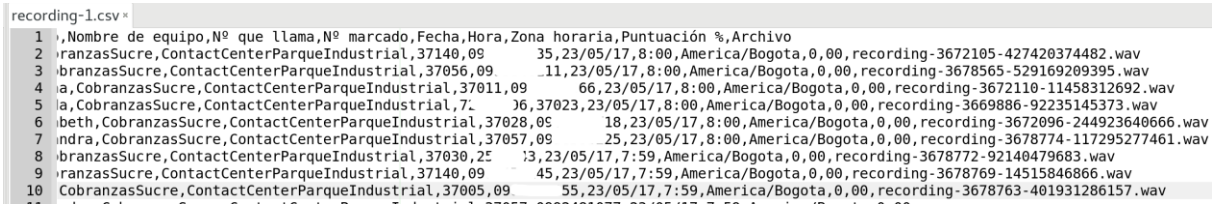
Ilustración 2 Listado de llamadas realizadas por el gestor de cobranzas en el sistema de callmanager Cisco

De la información descargada, el sistema genera un nombre de archivo único por cada archivo de audio como se puede apreciar en la ilustración:

Nombre	Tamaño	Modificado
recording-3643085-132379172370.wav	957.5 kB	15 may
recording-3643086-73926133965.wav	890.3 kB	15 may
recording-3643087-773024042779.wav	554.9 kB	15 may
recording-3643088-96663897695.wav	1.1 MB	15 may
recording-3669730-246376705999.wav	1.4 MB	23 may
recording-3669731-706525337843.wav	289.0 kB	23 may
recording-3669732-868502462758.wav	2.3 MB	23 may
recording-3669733-940362085018.wav	2.2 MB	23 may

Ilustración 3 Archivos de audio descargados desde el sistema de callmanager

Si bien los sistemas de callcenter tienen interfaces de integración para componentes de terceros, cada uno tiene su forma de implementación muy particular. Por este motivo se recurre a las grabaciones, ya que en cualquier sistema sería posible acceder a los ficheros de audio previamente grabados, y asociarlos al CDR (Call Detail Record) para saber a qué cliente se refiere cada uno.



recording-1.csv

```
1  ),Nombre de equipo,Nº que llama,Nº marcado,Fecha,Hora,Zona horaria,Puntuación %,Archivo
2  ranzasSucre,ContactCenterParqueIndustrial,37140,09      35,23/05/17,8:00,America/Bogota,0,00,recording-3672105-427420374482.wav
3  branzasSucre,ContactCenterParqueIndustrial,37056,09      11,23/05/17,8:00,America/Bogota,0,00,recording-3678565-529169209395.wav
4  a,CobranzasSucre,ContactCenterParqueIndustrial,37011,09      66,23/05/17,8:00,America/Bogota,0,00,recording-3672110-11458312692.wav
5  a,CobranzasSucre,ContactCenterParqueIndustrial,7     06,37023,23/05/17,8:00,America/Bogota,0,00,recording-3669886-92235145373.wav
6  beth,CobranzasSucre,ContactCenterParqueIndustrial,37028,09      18,23/05/17,8:00,America/Bogota,0,00,recording-3672096-244923640666.wav
7  ndra,CobranzasSucre,ContactCenterParqueIndustrial,37057,09      25,23/05/17,8:00,America/Bogota,0,00,recording-3678774-117295277461.wav
8  branzasSucre,ContactCenterParqueIndustrial,37030,25     3,23/05/17,7:59,America/Bogota,0,00,recording-3678772-92140479683.wav
9  ranzasSucre,ContactCenterParqueIndustrial,37140,09      45,23/05/17,7:59,America/Bogota,0,00,recording-3678769-14515846866.wav
10 CobranzasSucre,ContactCenterParqueIndustrial,37005,09      55,23/05/17,7:59,America/Bogota,0,00,recording-3678763-401931286157.wav
```

Ilustración 4 Formato de correspondencia de archivo de audio y registro de callmanager

El sistema del callcenter permite descargar un archivo de registros, que contiene la información que permitirá la asociación del archivo de audio con datos identificativos para que se pueda actualizar el resultado del clasificador. La estructura que tiene este archivo es el siguiente:

Campo	**Descripción**
Apellidos	Apellidos del gestor que realiza la llamada
Nombre	Nombre del gestor que realiza la llamada
Nombre del grupo	Grupo de trabajo del callcenter
Nombre del equipo	Ubicación del grupo
N.º que llama	Número de extensión desde el cual se contacta
N.º que marca	Número de teléfono al que llama
Fecha	Fecha de la llamada
Hora	Hora de la llamada
Zona horaria	Zona horaria
Puntuación %	Valor para puntuación (No se utiliza ya que no se puntúa)
Archivo	Nombre del archivo de audio de la grabación

Tabla 3 Formato de archivo de correspondencia

Basándose en el número telefónico marcado y el archivo del audio, se puede realizar la asociación para que el resultado del clasificador sea almacenado y visualizado por los gestores para que puedan identificar rápidamente al socio con quien van a conversar.

4.3. Preparación de los datos

En esta etapa de preparación de los datos, se realiza el proceso de selección del conjunto de datos que se utilizará para el análisis de acuerdo al tamaño de la nuestra previamente establecido. Para aplicar las técnicas de modelado seleccionadas se realizarán una serie de transformaciones sobre este conjunto de datos, transformaciones que pueden convertirse en un proceso cíclico en combinación con el modelado, con la finalidad de mejorar el algoritmo de clasificación alcanzando niveles más altos de precisión.

Se creará el corpus de respuestas de los socios que servirá de entrada para el proceso de anotación.

En la anotación se realiza un etiquetado basándose en un proceso predefinido sobre cada respuesta que se obtenga de un archivo de audio. Esta anotación es un proceso manual, realizado por un anotador humano, quien decide y asigna una etiqueta a nivel de respuesta con un criterio unificado. El resultado de esta anotación es un corpus con relevancia y polaridad que se utilizará en el modelado.

4.3.1. Transcripción

Consiste en el proceso de extracción de la información de un archivo de audio que contiene la grabación de una conversación establecida entre el gestor de cobranzas y el socio.

Los sistemas de callcenter manejan un estándar tradicional de archivos WAVE (WAV). Se trata de un formato de audio digital que no comprime los datos, el cual es comúnmente utilizado para almacenar sonidos en el sistema informático y contiene sonidos en dos canales, con una velocidad de muestreo de 8000 Hz. (WAVE Signed 16 bit Little Endian, Rate 8000 Hz, Stereo, 2 Channels) utilizando como códec de audio PCM (modulación por impulsos codificados).

Al tener dos canales, el primer canal contiene la información de las palabras que dijo el gestor de cobranzas del callcenter; el segundo canal, contiene la información de las palabras con las cuales respondió el socio que recibió la llamada.

4.3.2. Herramientas de reconocimiento de voz

Como se identificó en el estado del arte, existe un camino recorrido en el reconocimiento automático de voz, los sistemas de reconocimiento de voz han ido apareciendo, y ahora contamos con sistemas comerciales entre los cuales podemos citar: Google Speech API, Microsoft Bing Speech API, Nuance Recognizer, IBM Watson entre otros. Por otro lado, han aparecido sistemas no propietarios (open source) que tienen el propósito de superar y ofrecer un poco más de control sobre los modelos y el reconocimiento de características. Tradicionalmente, el estado de arte del reconocimiento de voz tuvo un enfoque basado en la fonética, por medio de la separación de componentes por pronunciación, acústica, y modelos de lenguaje. Estos se lograban con la combinación de modelos de lenguaje n-gram con modelos Hidden Markov (HMM). Hoy en día se ha introducido enfoques basados en RNN (Recurrent Neural Network), con algoritmos de deep learning con la finalidad de mejorar los modelos acústicos.

La mayoría de las herramientas abiertas o libres usan los modelos tradicionales basados en n-grams y HMM. Entre las más conocidas en el medio tenemos:

CMU Sphinx

Sistema de reconocimiento de voz desarrollado por la Carnegie Mellon University (Shmyrev, s.f.), el cual utiliza decodificadores de voz con modelos acústicos de ejemplo, además de programas necesarios para el entrenamiento de los modelos acústicos de lenguaje y un diccionario que se utiliza llamado cmudict, compuesto por una serie de versiones para propósitos específicos:

- Sphinx: sistema base que permite el conocimiento basado en modelos HMM y n-gram, interpreta voz hablada de forma continua.
- Sphinx 2: reconocimiento de voz en tiempo real utilizado para sistemas de aprendizaje de idiomas, además puede ser utilizado en los callcenters basados en Asterisk.
- Sphinx 3: procesamiento en batch con mejor precisión, acompañado de un programa de entrenamiento que utiliza técnicas modernas para mejorar la precisión.
- Sphinx 4: escrito en Java, con una completa reestructuración del motor, pensado para la independencia de la plataforma.
- PocketSphinx: diseñada para sistemas embebidos, como por ejemplo teléfonos inteligentes y equipos portátiles

Kaldi

Es un conjunto de herramientas para el reconocimiento de voz, pensado para ser flexible y extensible de una manera fácil, nace en la Universidad John Hopkins (Kaldi ASR, s.f.), y ha ganado una buena reputación rápidamente debido a la sencillez de trabajar con ella.

Soporta transformadas lineales, mutual information (MMI), modelos condicionales usados en Machine Learning; además de agregar DNN (Deep Neural Network) con 3 implementaciones:

- nnet1 provee soporte para entrenamiento utilizando una sola GPU, que permite que sea simple de entender y fácil de modificar.
- nnet2 provee soporte para entrenamiento más flexible, se puede utilizar múltiples GPUs y CPU, con soporte multithread.

- nnet3 una nueva implementación que permite utilizar redes neuronales más complejas que las redes feedforward. Las neuronas se organizan en distintas capas, de forma que las salidas de las neuronas de una capa se conectan a las entradas de varias o todas las neuronas de la capa superior. Permite utilizar múltiples GPUs en paralelo en varias máquinas.

Contiene un conjunto de ejemplos para realizar la construcción de un sistema de reconocimiento de habla, trabaja y ha sido entrenado con base de datos muy grande que provee Linguistic Data Consortium (LDC).

Fue pensado para resaltar el uso de algoritmos genéricos y recetas comunes, algoritmos simples de entender como transformadas lineales, en lugar de especificar algoritmos específicos al habla, facilitando el uso de cualquier origen de datos.

A continuación, se puede revisar un cuadro comparativo de estas dos herramientas tomando información de github:

Parámetro	CMUSphinx	Kaldi
Lenguaje	C, Java, Python, C++, CMake	C++
Likes o stars	1023	1790
Forks	246	915
Seguidores	118	258
Commits	1477	7303
Branchs	1	6
Release	0	0
Contribuyentes	1	125

Tabla 4 Comparación de motores de reconocimiento de voz open source

Además, existen otras herramientas como HTK, Julius e ISIP, las cuales no se han tomado en cuenta ya que en la documentación revisada y de acuerdo con la actividad en los repositorios, los proyectos Kaldi y CMU Sphinx con los que más actividad tienen.

Los dos proyectos han nacido como problemas de investigación académica, CMU Sphinx tiene más de 20 años y en el repositorio GitHub se visualiza un solo contribuyente. Esto no refleja la realidad de la historia del proyecto, ya que éste residía en SourceForge, en donde constaban 9 administradores y una docena de desarrolladores (CMUSphinx, 2017). Kaldi mientras tanto aparece en un workshop en el 2009, tiene 125 contribuyentes además un buen reconocimiento dentro de GitHub si tomamos en cuenta el número de forks y estrellas (Kaldi-ASR, 2017). Basándose en la actividad en la comunidad se dispone de listas de discusión y mailing en donde los desarrolladores se involucran. CMU Sphinx manejaba estos canales en SourceForge, pero al dar el paso a GitHub estos contenidos no han sido migrados, Kaldi por su lado mantiene estos canales activos en GitHub.

La documentación, tutoriales y ejemplos CMU Sphinx tiene un formato fácil de leer y seguir; Kaldi por su parte es un poco más difícil de seguir, aunque es comprensible. En Kaldi se hace referencia a la forma de aplicar los enfoques de aprendizaje basados en fonética y deep learning, mientras que los modelos CMU Sphinx vienen con modelos ya entrenados listos para usar, sobre todo en idioma inglés. Kaldi tiene modelos que son más difíciles de encontrar, incluso en inglés, que suelen utilizar datasets de VoxForge para entrenar los modelos.

El conocimiento que se tenga de los lenguajes de programación de cada herramienta puede influenciar cual utilizar. La mayoría utiliza Python para desarrollar el proyecto, por lo que es necesario contar con envolventes (wrappers) disponibles de manera oficial, o desarrollado como librería de un tercero, en cuyo caso no necesariamente se puede consumir todas las funcionalidades que tengan cada herramienta.

Gaida et al. (2014) en una comparación que realizan sobre las herramientas open source utilizan estos dos conjuntos de datos:

- Verbmobil 1 corpus (VM1): es un proyecto fundado por el Ministerio de ciencia y tecnología de Alemania, que se desarrolló entre los años 1993 y 2000, este conjunto de datos contiene conversaciones habladas en tres idiomas, inglés, japonés y alemán.

- Wall Street Journal 1 corpus (WSJ1): corresponde a lecturas habladas en inglés, realizadas en voz alta al leer noticias de Wall Street Journal, fue publicado en 1994.

Al comparar los tiempos que se utilizaron para configurar, preparar, correr y optimizar los motores de reconocimiento, expresan que se utilizó más tiempo en CMU Sphinx y menos para Kaldi. En la siguiente tabla se presentan los resultados por medio de la tasa de error de palabras (word error rates):

Herramienta	Porcentaje de error en VM1	Porcentaje de error en WSJ1
Sphinx-4	26,9	22,7
Kaldi	12.7	6.5

Tabla 5 Resultados medidos en WER (Gaida et al., 2014)

Se puede observar que Kaldi ofrece un buen resultado, proporciona un conjunto de herramientas incluidas, con un proceso lógico, ordenado y fluido desde el entrenamiento hasta la decodificación o transcripción y utiliza técnicas avanzadas con redes neuronales.

Como se mencionó previamente, existe variedad de herramientas comerciales, pero en este estudio se ha utilizado únicamente dos motores que exponen APIs públicas:

Google Speech API

Google posee una gran ventaja frente a muchos de sus competidores, debido a que tiene un conjunto de aplicaciones y servicios, que han sido utilizadas de fuentes de datos, para mejorar sus modelos, aumentando la precisión al transcribir un audio. Se puede citar servicios y aplicaciones como: búsqueda por voz en el escritorio por medio de Chrome, en dispositivos móviles por medio de Android, GOOG-411, transcripción y traducción en YouTube, entre otras.

Cuando Google empezó a adentrarse en tecnologías basadas en redes neuronales, adquirió compañías enfocadas en deep learning y con el paso de los años, otras como DeepMind, DNNresearch y Jetpac en donde han reforzado la investigación en machine learning. Ha registrado un 8% de error en el 2015 con una reducción del más del 23% desde el año 2013 (Speech API - Speech Recognition Google Cloud Platform, s.f.; Speech recognition, 2017).

Microsoft Bing Speech API

Microsoft empezó a desarrollar un API de habla desde 1993. Contrató a 3 de los 4 desarrolladores responsables de CMU Sphinx 2, con un desarrollo continuo para mejor e incrementar el poder de la plataforma del habla, incluyendo componentes en sus sistemas operativos. Esta mejora se basa en el uso de context dependent deep neural network hidden Markov model (CD-DNN-HMM) además de un vocabulario más grande.

Este motor convierte audios a textos, ya sea desde archivo o directamente desde un micrófono en tiempo real. Ofrece características de procesamiento en tiempo real: conforme se entrega el audio, se empieza a devolver una respuesta, la transcripción como tal. Al igual que Google, es una solución en la nube, que utiliza algoritmos avanzados con modelos entrenados provistos por el proyecto LUIS (Language Understanding Intelligent Service) (Bing Speech API, s.f.; Speech recognition, 2017).

4.3.2.1 Experimento de transcripción

Para efectos de este trabajo se utiliza un archivo de audio con una frase simple en ambos motores y obtenemos lo siguiente:

CMU Sphinx:

- Características del archivo de sonido: Rate 16000 Hz, Mono
- Texto dentro del audio: “hoy es feriado no se trabaja”

Para una prueba de concepto simple se utiliza un pequeño programa escrito en Python que hace uso de PocketSphinx, que es una librería tipo envolvente, que va a invocar al motor de reconocimiento de voz sphinx.

Como resultado de la ejecución del programa se obtiene el siguiente resultado:

“paul te será de si color se trabaja”

No se obtiene como resultado el texto que se citó previamente, lo único que coincide es “se trabaja”. Existe un gran porcentaje de error en las palabras reconocidas. Se utilizó el modelo de entrenamiento CMUSphinx Spanish Voxforge mismo que en la fase de entrenamiento consiguió los siguientes resultados: porcentaje total de aciertos = 74.72%, WER = 26.81% y exactitud = 73.19%

Kaldi

- Características del archivo de sonido: Rate 8000 Hz, Mono
- Texto dentro del audio: “the roof may not come down yet”

En este caso se utiliza un modelo pre entrenado. Al igual que en el caso anterior, en Kaldi no existe un modelo pre entrenado para español, por lo tanto, para la prueba de concepto se utilizó un modelo entrenado para inglés. Ya que los corpus utilizados por el motor le pertenecen a Linguistic Data Consortium, no son de fácil acceso.

Se realiza una prueba utilizando el conjunto de scripts de ejemplo que viene con Kaldi. En este caso, se utilizan los scripts que se encuentran en el directorio egs/aspire, los cuales se descargan de la página oficial de Kaldi, para entrenarlos y ajustar los parámetros necesarios.

Como resultado de la ejecución, la decodificación o transcripción del archivo del audio se obtiene el siguiente resultado:

“they may not comes on yet”

Se puede apreciar que no coincide el texto resultante con el texto original. En las pruebas con del modelo y los datos de entrenamiento utilizado se consiguió un 15% de palabras con error.

Google:

- Características del archivo de sonido: Rate 16000 Hz, Mono
- Texto dentro del audio: hoy es feriado no se trabaja

Para acceder y consumir este servicio se puede acceder de manera pública o por medio de una cuenta de Google Cloud en modo de pruebas. Para consumirlo se utiliza los clientes provistos por Google escritos en Python y disponibles en los repositorios públicos en GitHub.

Se modifica el código del programa transcribe.py para especificar el idioma al cual debe decodificar el archivo de audio para la prueba. Al ejecutar el programa sobre el archivo de archivo de audio se muestra la siguiente salida:

Transcript: hoy es feriado no se trabaja

Se puede observar que realizó una transcripción correcta del audio hacia texto con una confianza de 96% que retorna Google con el texto de la transcripción.

Microsoft:

- Características del archivo de sonido: Rate 16000 Hz, Mono
- Texto dentro del audio: hoy es feriado no se trabaja

Se ofrece una API puede ser consumida por medio de dos interfaces:

- REST API utilizado para procesamiento en lotes.

- Client Library utilizado para procesamiento en tiempo real, el cual se puede incluir en aplicaciones Android, iOS y Windows.

Para realizar la prueba se utiliza el cliente REST con un ejemplo que se ofrece en el mismo sitio de Microsoft, Bing Speech-to-Text Javascript SDK que está disponible en GitHub.

Se modifica el código del archivo index.html en la función getLanguage para especificar que el idioma al cual debe decodificar es al español, además de indicar que el archivo de audio sea el necesario. Luego de ejecutar el programa de ejemplo se obtiene el siguiente resultado:

[{"lexical":"hoy es feriado no se trabaja","display":"Hoy es feriado no se trabaja.","inverseNormalization":null,"maskedInverseNormalization":null,"transcript":"Hoy es feriado no se trabaja.","confidence":0.94373}]

Se puede observar que el servicio de Bing Speech API, al igual que el de Google, realiza una transcripción correcta del audio, pero la confianza o precisión es menor que la de Google, además de que Google contiene variaciones de los modelos para contemplar el español latinoamericano (Këpuska y Bohouta, 2017).

Con estos resultados se optó por escribir una aplicación cliente, que se convierte en el primer componente o subsistema del software, que utilice las librerías que entrega Google para consumir sus APIs por medio de Python. Debido a que en un archivo de audio existe información en los dos canales, es necesario dividirlo para cada canal, por lo tanto, al motor de reconocimiento de audio se le envía dos peticiones de decodificación o transcripción de audio a texto. El resultado de este proceso se almacena en una base de datos de la cual se extraerá los textos que son el insumo para crear el corpus.

El proceso de transcripción no es 100% confiable, como se demostró en la experimentación, ya que existen factores que alteran el audio tales como: el ruido, palabras propias de un zona o región en donde se encuentre el socio, el tipo de teléfono al que se le llamo: si es un fijo o móvil, entre otros, por lo que se tiene que contrastar los resultados con el mismo audio.

4.3.3. Pre procesado y creación del corpus inicial

Con la muestra establecida y la selección del motor de reconocimiento de voz, se estable una estructura de documento JSON con el formato clave, mismo que sirve para crear el corpus inicial. Este formato se describe de la siguiente manera:

{
"archivo": Nombre del archivo de audio,
"operador": Texto transcrito del canal 1 del archivo de audio,
"socio": Texto transcrito del canal 2 del archivo de audio,
"relevante": Especifica si el contenido del texto del campo o atributo socio contiene información relevante para el análisis de sentimiento,
"valoracion": Especifica si el contenido del texto del campo o atributo socio contiene una polaridad positiva o negativa
}

Se debe realizar procesos de validación y limpieza de los textos transcritos, ya que, al transcribir los audios, el texto resultante, no necesariamente contiene la respuesta como el socio la dijo. Este proceso, al igual que el anotado inicial, es manual.

4.3.4. Proceso de anotado

En el análisis de sentimientos el proceso de anotación consiste en identificar y clasificar la polaridad de las opiniones expresadas, en este caso las respuestas de los socios, que conforman el corpus formado previamente. La clasificación consiste en la asignación de etiquetas sobre cada respuesta identificada en el mismo.

El proceso de anotación es doble, ya que tiene que identificarse relevancia y luego polaridad, lo cual ayudará a que el modelo tenga información más filtrada para realizar la clasificación de la polaridad.

Sobre el corpus inicial se toma una pequeña parte de la muestra, con la finalidad de familiarizarse con las respuestas que se puede obtener de los diferentes tipos de socios, con los cuales el gestor de cobranzas (operador) estableció una conversación. De estas respuestas que ha dado el socio, se establece una pre clasificación con la finalidad de aclarar los criterios y consideraciones para convertirlos en una especie de guía y marco de referencia para la clasificación del todo el corpus.

Ejemplos de valoración:

A continuación, tenemos varios ejemplos de transcripciones textuales de conversaciones gestor – socio que han sido parte del estudio.

Valoración negativa

- Operador: Muy buenas noches disculpe la molestia con el señor Patricio Asmal buenas noches señor somos de la cooperativa JEP

  Socio: buenas noches si no le puedo ayudar en este momento llámame después de las 6 de la tarde

- Operador: Buenas noches me puede comunicar con la señora Ana Villa buenas noches somos de la cooperativa JEP le llamamos por un pago vencido de su crédito

  Socio: buenas noches no le puedo pagar porque no tengo plata ahora en este momento

Valoración positiva

- Operador: Buenas noches por favor me puede comunicar con el señor Sebastián Rivera si no se pusieron el pago para el día de hoy y mañana por favor haga lo que tiene el pago pendiente y amable Muchas gracias

  Socio: Buenas noches no está este momento es por el crédito Sí lo que pasa es que él no llega todavía del trabajo imagino que mañana le hace el depósito si si si él está pendiente de eso

- Operador: Buenas noches por favor el señor la señora María Josefina chang Alianza con quien tengo el gusto con quien tengo el gusto cómo le va señor somos En la cooperativa a dejar un recado con usted por favor indique la que tiene la cuota pendiente a cancelar En la cooperativa que nos ayude con el depósito ya listo Muchas gracias

  Socio: Buenas noches con quien tengo el gusto ya

- Operador: Muy buenas noches Disculpe la molestia por favor la señora Mariana Rosa rosado Cómo está mi señora Muy buenas noches Disculpe la molestia estoy llamando de la cooperativa JEP por favor me ayude indicándole a la señora Mariana Sofía delgado que por favor se acerque cancelar la cuota de un crédito que se encuentra pendiente ya mi señora Muchísimas gracias buenas noches

Socio: buenas noches y cómo está bueno bueno

Valoración no relevante

- Operador: muy buenas noches Disculpe la molestia con la señora Patricia de Los Ángeles Aguirre en noches

  Socio: buenas noches si espéreme un ratito

- Operador: Buenas noches con la señora Rocío Pinos disculpe la molestia le estamos llamando de la cooperativa JEP

  Socio: hola

En esta información se puede observar que existe una variedad de tipos de conversación que establece el gestor de callcenter con el socio al momento de realizar la llamada de gestión de cobranza o entregarle un mensaje. No todas las conversaciones son positivas o negativas, ya que existen conversaciones en donde no existen mayor interacción socio gestor porque responde el contestador automático, por lo tanto, no son relevantes para el modelo.

Una vez explicado los tipos de llamadas que podríamos encontrar, se procede a realizar una clasificación utilizando etiquetas como parte de la anotación de las respuestas, con los siguientes valores:

$$P = \{\text{relevante, no-relevante}\}$$

Se utiliza los siguientes criterios para el proceso de anotación a nivel de conversación:

- Si la respuesta del socio está completa, se entiende su significado y aporta suficiente información para determinar el nivel de aceptación ante la deuda pendiente, entonces se le asignará la etiqueta {relevante}.

- Si la respuesta no está completa, o no se entiende su significado, se le asignará la etiqueta {no-relevante}.

Etiqueta de relevancia

- No-relevante: esta etiqueta indica que la respuesta no agrega ninguna información. Puede ser consecuencia de que el socio una vez que escuchó que se le llama por una deuda pendiente no continuó la conversación, debido a errores de comunicación telefónica que provocaron que se pierda la señal y la conversación se interrumpió, o

dado a que el socio facilitó una repuesta fuera de contexto. A continuación, tenemos algunos ejemplos de respuestas textuales que entran en esta clasificación:

- buenas noches si espéreme un ratito
- buenas noches
- hola
- si buenas
- buen día
- buenas tardes
- buenos días
- aló

- Relevante: corresponde a toda respuesta que aporte información necesaria para determinar la polaridad de la misma. Algunos ejemplos textuales de esta clase son los siguientes:
  - buenas noches si no le puedo ayudar en este momento llámame después de las 6 de la tarde
  - buenas si no le puedo ayudar en este momento
  - buenas noches no le puedo pagar
  - aló no le voy a pagar
  - si ya no se preocupe yo le aviso
  - buenas tardes si está pendiente del pago
  - buenos días mañana le deposita
  - buenos noches ya listo
  - buenos noches si listo

Una vez filtradas las conversaciones que contienen información relevante para el análisis, se procede a realizar la anotación de las respuestas para determinar la polaridad que tuvo el socio.

Polaridad de respuestas

En esta fase se va a realizar un etiquetado sobre la respuesta del socio como una unidad, únicamente de las respuestas relevantes que contiene información para asignar una polaridad.

Teniendo claro que el objetivo principal de este trabajo es detectar las respuestas positivas y negativas, se utiliza el siguiente conjunto de etiquetas:

P = {negativa, positiva}

Se utiliza los siguientes criterios para el proceso de anotación a nivel de respuesta del socio:

- Si contiene alguna referencia negativa, es decir, que el socio no expresa la voluntad de pagar o de atender la llamada en este momento, en cuyo caso se le asignará la etiqueta {negativa}.

  Ejemplo de respuestas negativas:

  - buenas noches si no le puedo ayudar en este momento llámame después de las 6 de la tarde
  - buenas si no le puedo ayudar en este momento
  - buenas noches no le puedo pagar
  - aló no le voy a pagar
  - buenas días sabe que no tengo dinero así que no le voy a pagar

- Si, al contrario, contiene referencias positivas, es decir, el socio tiene la voluntad de pagar, de entregar un mensaje en el caso de que la persona involucrada no esté disponible, entrega un compromiso de pago o acepta el mensaje que le da el gestor de una manera objetiva, entonces se le asignará la etiqueta {positiva}.

  Ejemplo de respuestas positivas:

  - si ya no se preocupe yo le aviso

- buenas tardes si está pendiente del pago
- buenos días mañana le deposita
- buenos noches ya listo
- buenos noches si listo
- si ya mañana le deposita
- buenas noches si ya le hace una transferencia

De esta manera, posterior a la anotación manual, obtenemos el corpus inicial necesario para el modelado. Se almacenará en una base de datos persistente para manipulaciones y análisis futuros. Este corpus se lo ha representado como un arreglo de documentos JSON de la siguiente manera:

conversaciones.json × conversaciones-ok.json ×


```
[
  {
    "archivo": "recording-3643088-96663897695.wav",
    "operador": "Buenas noches por favor me puede comuni
    "relevante": 1,
    "socio": "aló de parte de quien lo que pasa es que él
    "valoracion": 1
  },
  {
    "archivo": "recording-3669730-246376705999.wav",
    "operador": "buenos días para mi hijo la señorita Mó
    "relevante": 1,
    "socio": "Buenos días si lo pasa es que estoy esperar
    "valoracion": 1
  },
  {
    "archivo": "recording-3669731-706525337843.wav",
    "operador": null,
    "relevante": 0,
    "socio": "Hola No estamos disponibles ahora ",
    "valoracion": null
  },
```


Ilustración 5 Arreglo de documentos con formato JSON

Como resultado de este proceso y de acuerdo con la estimación de la muestra previamente calculada, se obtiene un corpus con las siguientes características:

Numero de registros del corpus inicial	353
Respuestas con información relevante	294
Respuesta con información no relevante	59
Respuesta con valoración positiva	217
Respuestas con valoración negativa	77

Tabla 6 Tamaño del corpus inicial

4.4. Modelado

En el proceso de modelado se construye los clasificadores, los mismos que conforman los subsistemas principales del software:

- Clasificador de relevancia
- Clasificador de polaridad

Debido a que se trata de un problema de machine learning, de tipo clasificación, se debe seleccionar las técnicas que se enfocarán en la clasificación binaria. De acuerdo con la documentación existente, las técnicas de clasificación comúnmente empleadas son Naives Bayes y Support Vector Machines (SVM). Sin embargo, se puede emplear otras técnicas que se adapten o resuelvan problemas de árboles de decisión, regresión logística, redes neuronales entre otras.

Los clasificadores utilizan modelos matemáticos y estadísticos para resolver el problema. Entonces es necesario realizar una caracterización de los textos, mismo que consiste en definir un conjunto de variables independientes o características (features) en función de las necesidades que tiene cada clasificador. Posterior a la caracterización, se pueden aplicar los clasificadores en el conjunto de datos de entrenamiento. Con la finalidad de mejorar el modelo, se realiza un proceso cíclico, dependiendo de los resultados obtenidos. Se emplea validación cruzada y otros clasificadores para comparar los resultados.

Navie Bayes

Los modelos bayesianos se utilizan para resolver problemas tanto desde el punto de vista descriptivo como del predictivo. Desde la perspectiva descriptiva, se centra en descubrir las

relaciones de dependencia o independencia y se complementan con las reglas de asociación; desde la perspectiva predictiva, son utilizados como métodos de clasificación.

Los métodos bayesianos se han aplicado en el campo de la inteligencia artificial, en el aprendizaje automático y en la minería de datos por las siguientes razones:

- Constituyen métodos válidos, prácticos para realizar inferencias con los datos de entrenamiento, basados en modelos probabilísticos.
- Muy útiles en la comprensión de otras técnicas de inteligencia artificial y minería de datos que no trabajan con probabilidades, ya que al combinarlas con las técnicas bayesianas se optimizan las soluciones a los problemas planteados.

Los modelos o técnicas bayesianas se basan en el teorema de Bayes descrito por la siguiente fórmula (Keller, 2002):

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)}$$

Donde:

- P(h) es la probabilidad a priori de que se cumpla la hipótesis h.
- P(D) es la probabilidad de D.
- P(D|h) es la probabilidad condicional de D dado h.
- P(h|D) es la probabilidad a posteriori, probabilidad condicional de h dado D (Keller, 2002).

Al hablar de probabilidad, el siguiente paso es conseguir la hipótesis más probable o hipótesis MAP (maximum a posteriori) con la siguiente expresión (Keller, 2002):

$$h_{map} = argmax_{h \in H} P(D \mid h)P(h)$$

hmap es la hipótesis más probable dados los datos observados P(h|D)

En los problemas de clasificación se tiene un conjunto de entrenamiento formado por instancias que tienen un conjunto de atributos, cuyo objetivo es determinar la clase, misma que tiene un conjunto de valores finitos v, de tal manera que se pueda predecir la clase correcta para las nuevas instancias a_1, a_2, …, a_n (Keller, 2002).

$$v_{map} = argmax_{v_j \in V} P(a_1, a_2, \ldots a_n \mid v_j)P(v_j)$$

Naive Bayes parte de la suposición de que todos los atributos son independientes entre sí, con respecto a la clase, categoría o concepto. Es uno de los métodos más competitivos comparados con otras técnicas como las redes neuronales o árboles de clasificación (Kononenko, 1990).

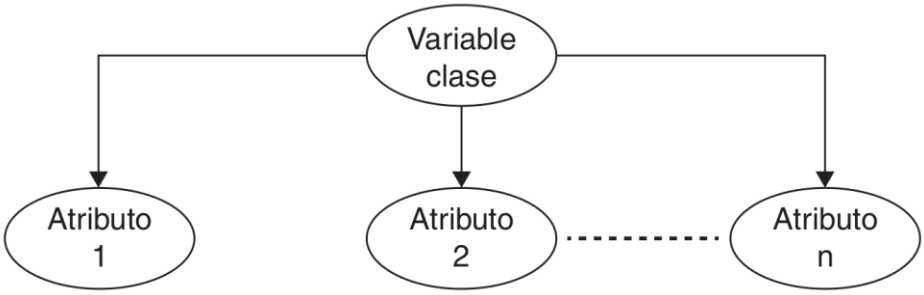


Ilustración 6 Estructura Naive Bayes

La estimación de los parámetros para este método, la clase o valor a devolver será el resultado de aplicar la siguiente fórmula (Keller, 2002):

$$v_{NB} = argmax_{v_j \in V} P(v_j) \Pi_i P(a_i \mid v_j)$$

Entre las propiedades del teorema de Bayes tenemos:

- Con cada instancia o ejemplo de entrenamiento, el conocimiento previo y la probabilidad pueden ser actualizados dinámicamente haciéndolo flexible y robusto a errores.
- Combina el conocimiento a priori y los datos observados.
- No solo produce una clasificación, también una distribución de probabilidad sobre todas las clases.
- Se puede combinar la salida de varios clasificadores, por medio de la multiplicación de las probabilidades de todos los clasificadores para la misma clase (Keller, 2002).

Las aplicaciones prácticas en las cuales se ha destacado el uso de los teoremas de Bayes son el diagnóstico y la clasificación de texto. En el campo de la clasificación de textos, las instancias son documentos de texto, de donde se quiere aprender conceptos mediante el uso de ejemplos, como es el caso de los artículos que son de interés o no, páginas web que tienen un tema determinado o una categoría, entre otros.

Support Vector Machine

Las máquinas de vectores de soporte (SVM) fueron introducidas en los años 90 por Vapnik, con la finalidad de resolver problemas de reconocimiento de patrones basados en aprendizaje estadístico. Se lo creó resolver problemas de clasificación binaria y se lo ha extendido para problemas de regresión, agrupamiento, clasificación múltiple, detección de valores atípicos con aplicación práctica en el área de procesamiento de lenguaje natural, caracterización de textos, reconocimiento de caracteres, entre otros (Cortes y Vapnik, 1995).

Es un clasificador lineal y establece separadores lineales o hiperplanos, en el espacio original de los instancias o ejemplos de entrada, o en un espacio transformado.

Su objetivo principal es el de seleccionar un hiperplano de separación, que equidiste de las instancias cercanas de cada clase, para conseguir un margen máximo a cada lado del hiperplano. En esta selección, solo se consideran las instancias de entrenamiento de cada clase que caen justo en la frontera de los márgenes máximos. Estas instancias reciben el nombre de vectores de soporte (Suárez, 2014).

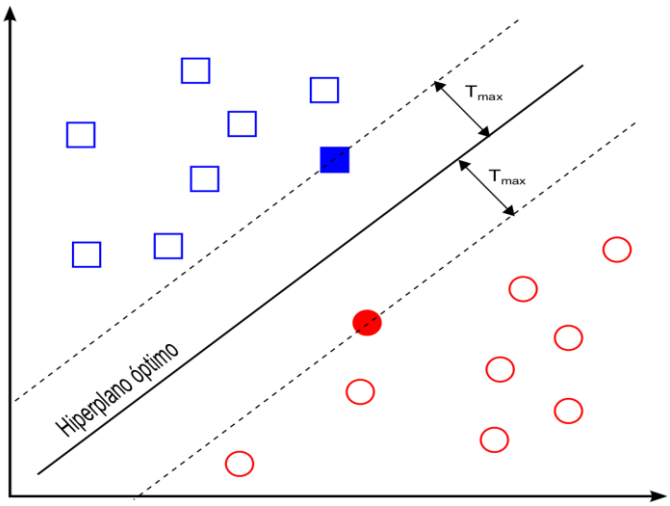


Ilustración 7 "Hiperplano de separación óptimo y su margen asociado (máximo)" (Suárez, 2014)

En la ilustración se puede apreciar una clasificación ideal (perfectamente lineal), en donde se aprecia claramente el mejor hiperplano de separación óptimo y el margen máximo asociado. Sin embargo, es un poco alejado de la realidad, con poco interés práctico, ya que los problemas de clasificación reales tienen instancias o ejemplos con ruido y no son perfectos ni linealmente separables. En este tipo de problemas reales la estrategia es relajar el grado de separabilidad del conjunto de instancias, admitiendo que exista errores de clasificación en algunos de las instancias del conjunto de entrenamiento, y así poder

encontrar un hiperplano óptimo para el resto de ejemplos que sí son separables (Suárez, 2014).

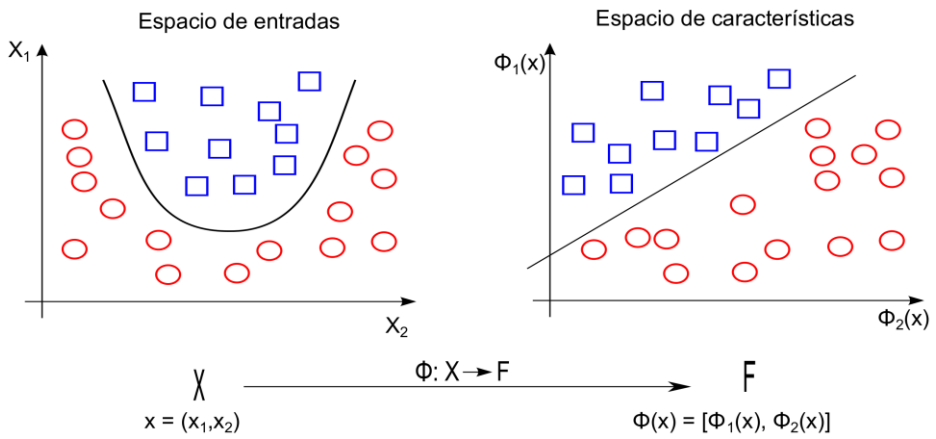


Ilustración 8 "Función de decisión no lineal en el espacio del conjunto de ejemplos original a un espacio transformado" (Suárez, 2014)

En la figura anterior se puede observar el problema de búsqueda de una función de decisión no lineal en el espacio del conjunto de ejemplos originales (espacio de entradas), que por medio de transformaciones se convierte en un nuevo problema consistente en la búsqueda de una función de decisión lineal (hiperplano) en un nuevo espacio de características (Suárez, 2014).

Las ventajas del Support Vector Machine son:

- Proceso de entrenamiento relativamente fácil, en comparación con las redes neuronales.
- Efectivo con espacios altamente dimensionales.
- Aún efectivo en el caso donde el número de dimensiones es más grande que el número de instancias.
- El compromiso entre la complejidad del clasificador y el error puede ser controlado explícitamente.
- Se pueden utilizar como datos de entrada para el SVM no sólo vectores de características, sino también cadenas de caracteres y árboles.

Entre las desventajas se tienen las siguientes:

- Si el número de características es mucho más grande que el número de ejemplos, el clasificador probablemente tendrá un bajo rendimiento.

- Se necesitan metodologías eficientes para establecer los parámetros de inicialización del SVM.

Support Vector Machine ha demostrado tener aplicaciones con gran desempeño, más que las redes neuronales en máquinas de aprendizaje tradicional, con muy buen rendimiento de problemas de clasificación, sobre todo cuando se tienen dos clases.

4.4.1 Clasificador de relevancia

El objetivo de este proceso es clasificar las respuestas del socio en relevantes y no relevantes utilizando técnicas de aprendizaje automático supervisado, seleccionadas a partir del corpus inicial ya anotado. Básicamente es una clasificación binaria.

El mayor reto de este clasificador es que las clases objetivo no se encuentra balanceadas, existe un buen desequilibrio. En la clase relevante tenemos 259 instancias que representan un 83.28% muy por encima de la mitad de las instancias; en cambio en la clase no relevante tenemos un 16.72% formado por 59 instancias de la muestra inicial.

Como se citó previamente, las técnicas seleccionadas para el modelado fueron Naive Bayes y SVM, debido a que entregan resultados de una manera simple y rápida. Para la experimentación de estos modelos iniciales se han utilizado las librerías NLTK (Natural Language Toolkit, s.f.) y Scikit-learn (Scikit-learn, s.f.) de Python, empezando por LinearSVC. Sin embargo, tras los primeros experimentos y resultados se decidió experimentar con otros clasificadores para comparar los resultados. Estos clasificadores fueron: Random Forest y Multi-layer Perceptron.

En los escenarios de prueba se utiliza parámetros de configuración propios de los algoritmos. Para encontrar los parámetros más adecuados se hace una búsqueda exhaustiva por medio del GridSearch; además se realiza una validación cruzada de los datos de la muestra, separando los datos en un conjunto para entrenamiento y un conjunto de datos para pruebas.

Extracción de características (feature extraction)

El primer paso del modelado es la extracción de información de los textos, para crear un conjunto de características con las que se alimenta al clasificador seleccionado.

La forma más común de hacerlo es transformar cada texto en un vector de palabras o términos, con los que se determinan características del conjunto de datos. De este conjunto

de características se eligen las mejores por medio de combinaciones de variables para incrementar el rendimiento de los modelos.

Vector de palabras

Es uno de los métodos ampliamente conocidos por ser fácil de implementar, ya que permite la extracción de características a partir del texto original de una opinión. El vector de palabras (Bag of Words, BOW), especifica que cada documento o texto *t* se define de la forma:

$$t = (W_1, W_2, W_3, \ldots, W_{|V|})$$

Donde $|V|$ es el tamaño del vocabulario total que tiene el corpus y cada W_i toma el valor de 0 o 1 si el término o palabra aparece en el texto *t.*

Con esta premisa, el primer paso es separar cada texto, en este caso cada respuesta del socio, en su vector de palabras correspondiente, proceso que se lo conoce como word tokenization.

De esta manera, el corpus de entrada con un tamaño *M* se transforma en una matriz de *N x M*, donde *N* representará el número de características (features) y *M* es el número de instancias u observaciones.

Por ejemplo, si elaboramos la matriz para los siguientes textos:

- “buenos días si con el mismo”
- “buenos días si dígame”

	buenos	días	si	con	el	mismo	dígame
Frase 1	1	1	1	1	1	1	0
Frase 2	1	1	1	0	0	0	1

Tabla 7 Ejemplo de vector de palabras

Para caracterizar los textos, la manera más simple es el uso del vector de palabras que toma valores binarios, en función de si el término aparece o no en el texto. En esta forma de caracterizar no se toma en cuenta la importancia de los términos, ya que puede existir términos diferentes dentro de cada texto. La aproximación más simple para determinar la

importancia de cada término es calcular la frecuencia de aparición de cada término (Term Frecuency, TF)

Por ejemplo, el cálculo del vector de palabras con término de frecuencia es la frase:

- “buenas noches si con el mismo si listo buenas noches”

Término	buenas	noches	si	con	el	mismo	listo
Frecuencia	2	2	1	1	1	1	1

Tabla 8 Ejemplo de vector de palabras con TF

Estas caracterizaciones tienen un problema que se deriva de las particularidades de nuestro idioma, en español las oraciones contienen artículos definidos (el, la, los, las), artículos indefinidos (un una unos unas), preposiciones (de, con, a, en, para), entre otras complejas reglas gramaticales. A estas palabras se las considera como stop words, ya que se convierten en una fuente de ruido para la construcción de los modelos, porque casi siempre son las que tienen mayor número de apariciones sin ser las más importantes.

En las pruebas se evaluaron las dos técnicas de caracterización para comparar los resultados y quedarse con los mejores valores, eliminando las stop words, evaluando la caracterización por medio de vector de palabras, así como las frecuencias de términos, y también aplicando la inversa de la frecuencia de aparición del término en el corpus.

La elección de la caracterización óptima se realiza en función del clasificador que se vaya a utilizar para el modelado, en donde influye obviamente el conjunto de datos a analizar, es decir el corpus inicial. Para este caso las respuestas son textos relativamente cortos (longitud máxima de 315 caracteres y mínima de 3 caracteres) y el número de instancias en el corpus es de 353, por lo cual se utilizó el método simple: el término aparece o no en el texto.

N-gramas

La técnica de caracterización por medio del vector de palabras es sencilla y ha ofrecido buenos resultados. Para mejorarlos se puede utilizar la generalización a N-gramas, en lugar de capturar información relativa al orden de las palabras.

N hace referencia al número de términos consecutivos, de esta forma si N = 2 se construyen bigramas, se utiliza la siguiente frase para contextualizar esta técnica:

- “buenas noches si con el mismo si listo buenas noches”

Los bigramas se pueden construir agrupando de la siguiente manera:

“buenos días” “si con” “el mismo” “si listo” “buenas noches” “días si” “con el” “mismo si”

De la misma manera, si N = 3 se construyen trigramas, y si N = 1 estamos hablando de unigramas, que es lo mismo que el vector de palabras.

Pruebas

En los algoritmos de los clasificadores utilizados en las pruebas, así como en la mayoría de algoritmos de machine learning, se presenta un problema de sobre entrenamiento produciendo un sobre ajuste (overfitting), debido a que los modelos se ajustan a las características muy específicas de los datos de entrenamiento reduciendo la capacidad de generalización que debe tener el algoritmo. Para mitigar y reducir lo más posible este factor, se utiliza la validación cruzada, donde no solo se divide el conjunto de datos en datos de entrenamiento y de prueba, sino que se realiza un análisis estadístico para obtener medidas de rendimiento estimado para entender como el rendimiento varía a través de los distintos conjuntos de datos.

Debido a que el conjunto de datos del corpus no es muy extenso, se utilizó una validación cruzada de 3 iteraciones, valor predeterminado en la búsqueda exhaustiva que se hace por medio de GridSearchCV que ayuda a encontrar los mejores parámetros para comparar los modelos obtenidos de acuerdo con la métrica AUC.

Con la finalidad de eliminar el ruido del espacio de características se aplica lo siguiente:

- Eliminación de palabras comunes (stop words) de las listas de características. Estas palabras no poseen una carga léxica relevante, por lo que aportan muy poca información a los modelos. Por lo tanto, se define una lista de palabras que pueden ser utilizadas en función de cada problema y dominio con la finalidad de mejorar los modelos. En este caso se utilizó la lista de palabras comunes que incluye la librería NLTK para simplificar el proceso.

- Reducción de los términos a su raíz (stemming). Se trata de agrupar términos que pueden ser derivaciones de la misma palabra, o palabras muy relacionadas semánticamente, ayudando a reducir el ruido del conjunto de datos. Para esto se utilizó la librería Snowball de NTLK.

En resumen, el proceso de modelado se compone de las siguientes etapas:

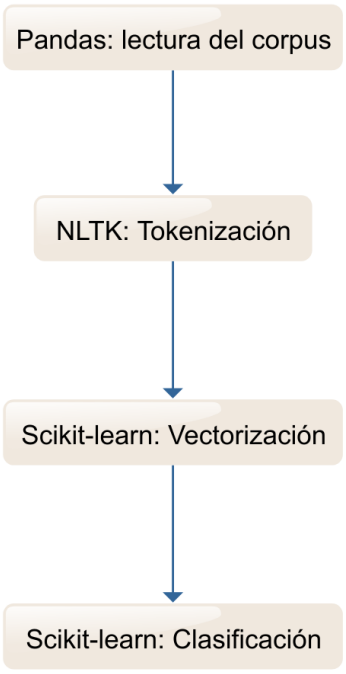


Ilustración 9 Pipeline de procesamiento

Se realiza pruebas para seleccionar las mejores características que ayuden a mejorar los resultados del modelo, empezando por determinar si se utilizará vector de palabras con términos de frecuencia o no.

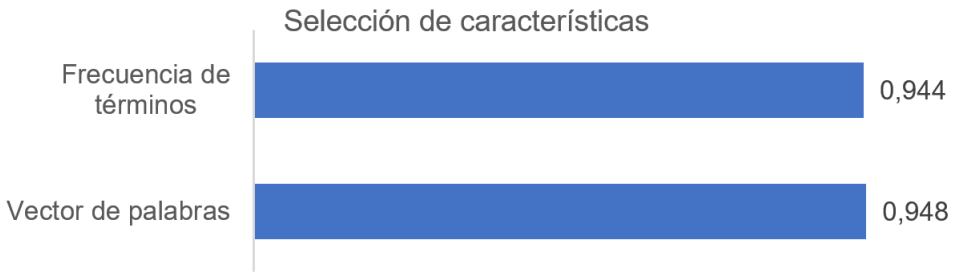


Ilustración 10 Comparación de vector de palabras con términos de frecuencia

El método que mejor resultado promedio aportó a los algoritmos, para la clasificación de relevancia, fue el vector de palabras. Si bien no existe una gran diferencia, influye de manera positiva en la métrica de calificación. Otro factor que se consideró fue el tiempo de procesamiento que se emplea para extraer las características.

Se realizó una comparación para determinar si dentro del vector de palabras, cada término individual contiene suficiente influencia, o si la combinación de dos términos agregaba más información al modelo.

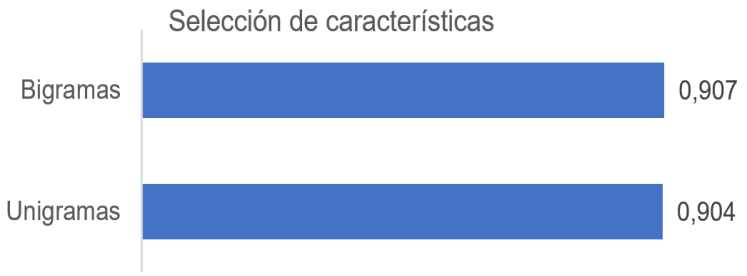


Ilustración 11 Comparación Unigramas vs Bigramas

Se pudo observar que el uso de bigramas para vincular dos términos consecutivos entregó una ligera mejora a los modelos de clasificación.

Luego de esto, se realizó pruebas para determinar si las palabras comunes (stop words) aportan información que sea relevante para los clasificadores o no, ya que en nuestro idioma son bastante utilizadas.

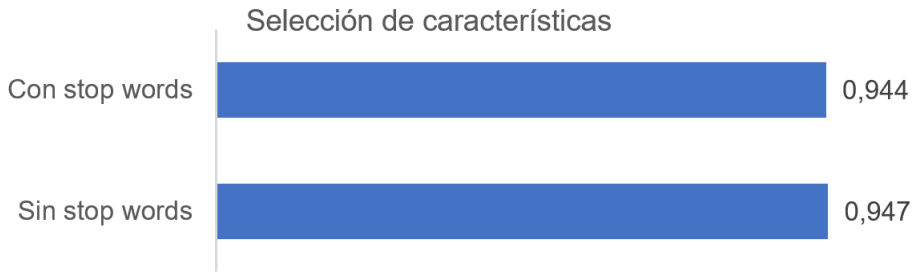


Ilustración 12 Comparación del uso o no de stop words

Basándose en la figura anterior, se puede observar que al eliminar los stop words del conjunto de términos, mejora ligeramente el promedio de los resultados de los clasificadores. Estas palabras comunes se las puede visualizar mejor en la siguiente nube de palabras que fue creada del corpus inicial, donde el tamaño de la fuente representa la frecuencia de aparición de la palabra.

Ilustración 13 Palabras comunes utilizadas con stop words

Claramente se pueden identificar en el gráfico anterior, que las palabras: ya, que, el, de, son las palabras comunes con mayor presencia. Eliminado las stop words, la nube de palabras tiene la siguiente forma:

Ilustración 14: Palabras comúnmente utilizadas sin stop words

En esta ilustración, se puede identificar de manera sencilla cuales son las palabras que tienen mayor relevancia para el dominio del problema, por ejemplo, los términos “aló” o “si”, que son generalmente las primeras palabras en la respuesta a una llamada.

Con la finalidad de reducir el ruido que puede presentarse por el uso frecuente de diminutivos (especialmente en ciertos modismos propios de las personas de una región) en términos como señorita, jovencita, ratito, entre otros, se realiza el proceso de stemming determinando la raíz de la palabra.

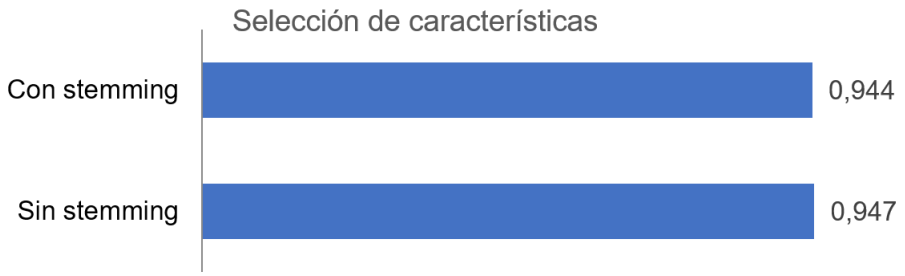


Ilustración 15 Comparación del uso o no de stemming

Después de realizar este proceso, se comprobó que la diferencia era muy corta. Esta ligera variación en los experimentos nos demostró de manera clara y contundente que no existe una mejora significativa al utilizar los procesos antes mencionados. Sobre este conjunto de datos pequeño que se tiene, no tienen gran influencia en los resultados. Únicamente, se los utiliza ya que son tareas comunes que forman parte un proceso de minería de textos.

Como última tarea del proceso, se realizó la comparación de los clasificadores. Se compararon los modelos basados en SVM y Naive Bayes, además de incluir estos dos clasificadores con la finalidad de tener más valores de referencia en la comparación:

- RandomForestClassifier: Pertenece a la familia de los algoritmos de clasificadores de árboles de decisión. Suelen tener buenos resultados para problemas diversos, ya que controla muy bien el sobreajuste y proporciona muy buena exactitud. En los problemas donde el conjunto de datos tiene una alta dimensionalidad se ve afectado significativamente, ya que los subespacios de características que crea son aleatorios para cada árbol, llegando a incluir características que no tienen relevancia.

- MLPClassifier: Es un clasificador que utiliza redes neuronales artificiales, multilayer perceptrón. Consiste en al menos 3 capas de nodos y es utilizado para resolver problemas de aprendizaje supervisado de clasificación o regresión, asociación de patrones, segmentación de imágenes, comprensión de datos, etc. Tiene desventajas que están asociadas a la existencia de varios mínimos locales además de problemas de ajuste de los valores apropiados para los parámetros de iteraciones, capas y neuronas.

Estos clasificadores se escogieron de acuerdo a la guía que entrega la herramienta Scikit learn utilizada en el modelado.

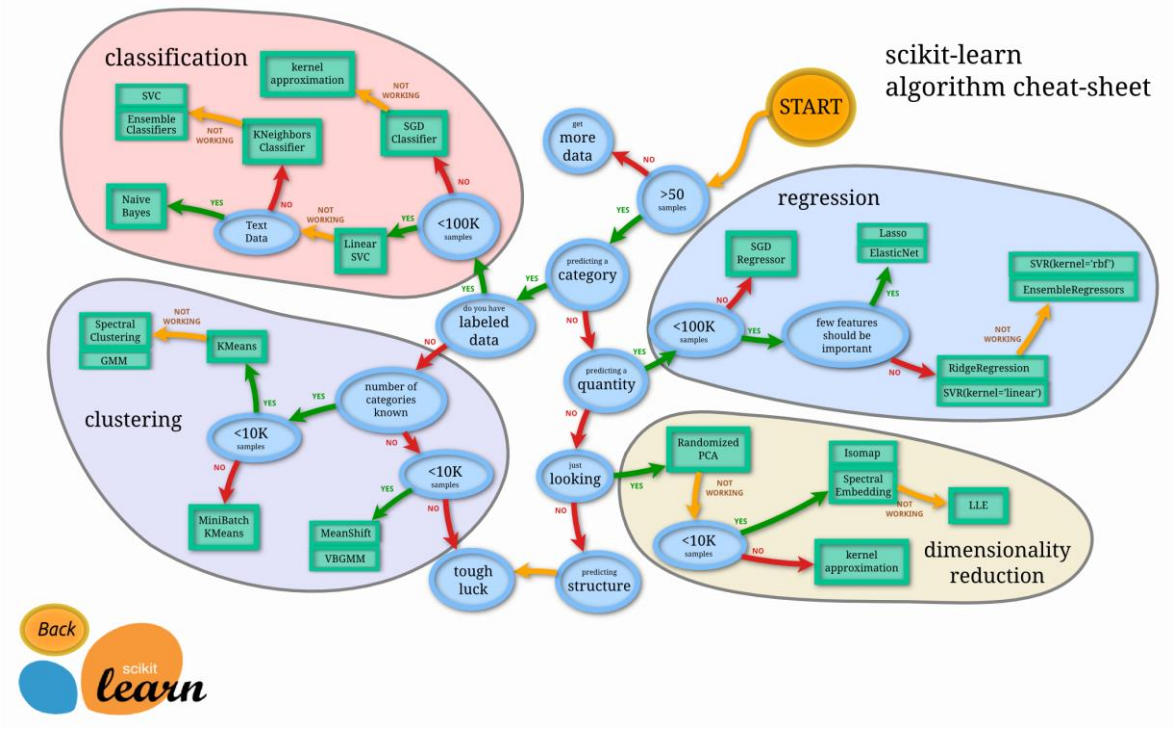


Ilustración 16 Guía de selección de algoritmos de machine learning (Scikit-learn, s.f.)

A estos dos últimos clasificadores se les aplicó la misma validación sobre el mismo conjunto de datos de entrenamiento.

Para las pruebas de los clasificadores se utiliza un Pipeline que ofrece Scikit-learn, que es una lista secuencial de tareas o pasos que se interconectan siguiendo un orden lógico y permite asignar parámetros propios en cada uno de estos pasos. Por ejemplo, el pipeline que se utilizó para el clasificador SVM fue el siguiente (Scikit-learn, s.f.):

```
pipeline = Pipeline([
('vect', vectorizer),
('cls', LinearSVC()),
])
```

Se conectaron de esta manera los procesos de tokenización y el proceso de clasificación.

Además, se realizó una búsqueda exhaustiva por medio de GridSerchCV que es un enfoque para realizar un afinamiento de los parámetros. Obviamente los valores de los parámetros dependen de cada clasificador y los posibles valores que pueden tomar dependen de cómo influyen estos valores en el dominio del problema. GridSearchCV realiza combinaciones sobre los valores que se asignó a los parámetros sin la necesidad de escribir bucles o

iteraciones en el código, y así con los resultados realizar las operaciones de cálculo para identificar el mejor juego de parámetros. Por ejemplo, para determinar los mejores parámetros del clasificador SVM se utiliza el siguiente pedazo de código basado en la documentación oficial (Scikit-learn, s.f.):

```
parameters = {
'vect__max_df': (0.5, 1.9),
'vect__min_df': (10, 20, 50),
'vect__max_features': (100, 500),
'vect__ngram_range': ((1, 1), (1, 2)),
'cls__C': (0.2, 0.5, 0.7),
'cls__loss': ('hinge', 'squared_hinge'),
'cls__max_iter': (500, 1000)
}
grid_search = GridSearchCV(pipeline, parameters, n_jobs=-1 ,
scoring='roc_auc')
```

A este conjunto de parámetros se le asigna el nombre Test 1, se realiza la búsqueda exhaustiva con el estimador AUC para su comparación. Estos conjuntos de parámetros se reutilizan para el resto de clasificadores, excepto los que son propios del clasificador.

Se modifica el valor de los parámetros de Test 1 para obtener nuevos resultados con los cuales se comparará con los valores de AUC.

Continuando con el ejemplo, el conjunto de parámetros se modifica de la siguiente manera:

```
parameters = {
    'vect__max_df': (0.5, 1.0, 1.5, 2.0),
    'vect__min_df': (1, 5, 10, 20, 50),
    'vect__max_features': (100, 250, 500, 1000),
    'vect__ngram_range': ((1, 1), (1, 2)),
    'cls__C': (0.1, 0.5, 0.9),
    'cls__loss': ('hinge', 'squared_hinge'),
    'cls__max_iter': (500, 1000)
}
```

Este nuevo conjunto de parámetros se lo llamará Test 2. Con este nuevo conjunto se repite la búsqueda exhaustiva para cada uno de los clasificadores.

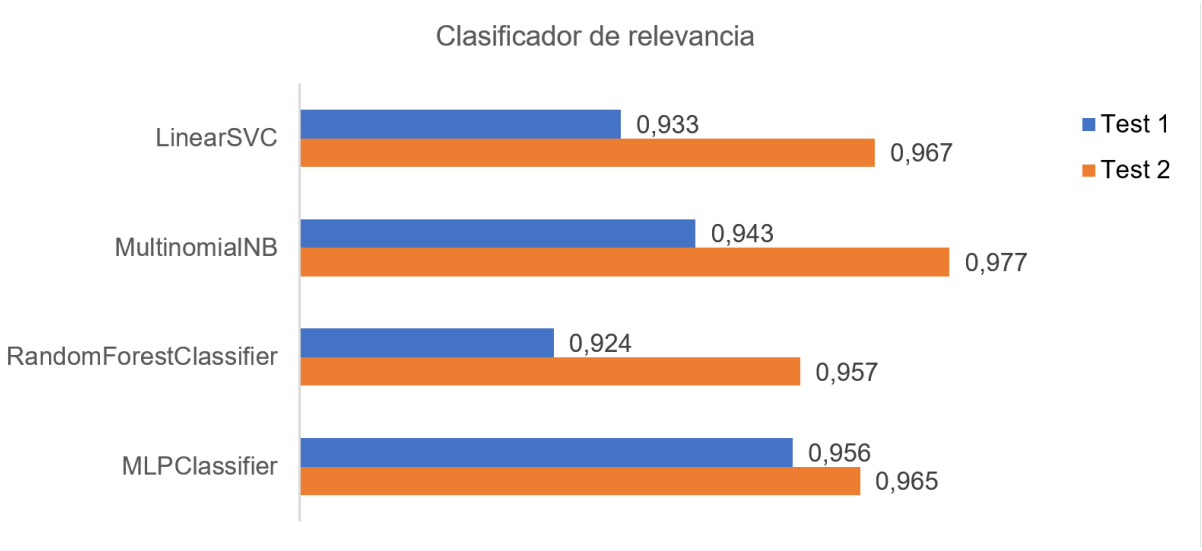


Ilustración 17 Comparación de resultados de los algoritmos de clasificación basado en AUC

Basándose en los resultados de los clasificadores usados, se puede observar que las medias obtenidas están todas por encima del 0.90 de AUC, destacando el clasificador MultinomialNB que llega a obtener en el conjunto de parámetros de Test 2, el valor más alto de todos los clasificadores con una media de 0.97 de AUC.

Realizar la búsqueda exhaustiva es iterar sobre el conjunto de parámetros de Test. Existe un costo computacional ligado a estas iteraciones. La métrica más representativa de este costo es el tiempo, en el que el clasificador llego a entregar los resultados en las diferentes combinaciones.

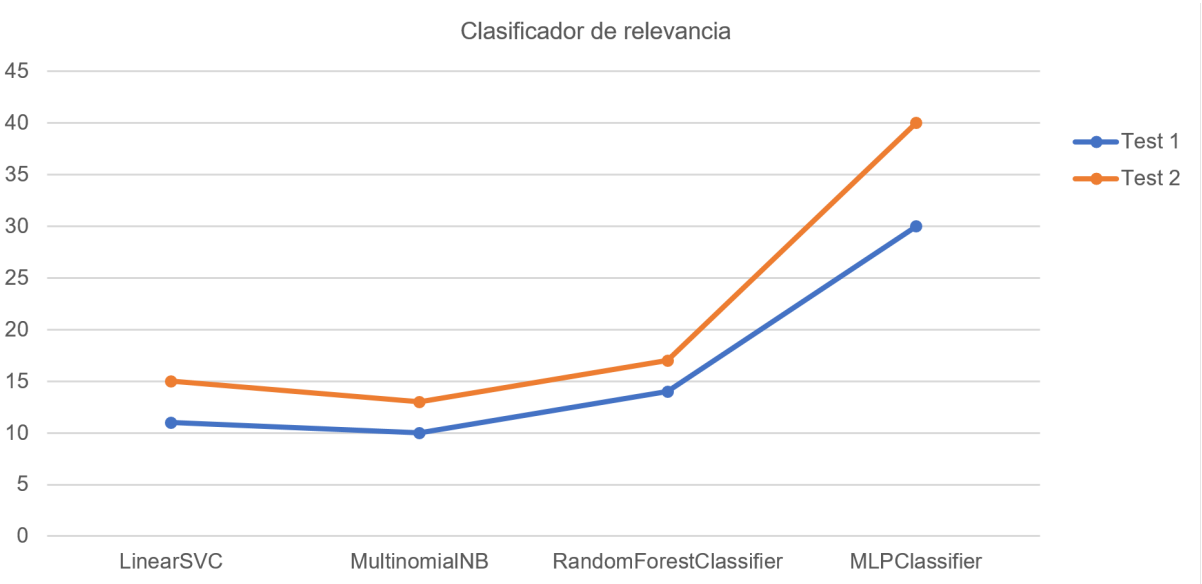


Ilustración 18 Comparación de los tiempos tomados de los algoritmos de clasificación

Los tiempos que se visualizan están, no solo determinados por el clasificador, sino que dependen de las capacidades del ordenador en donde se corrieron los test. Se puede observar que los rangos de tiempos se encuentran entre los 10 y 15 minutos, con un incremento muy notable en los test de redes neuronales, en donde se duplican los tiempos manejados por los otros clasificadores.

En el modelo de clasificación de relevancia de acuerdo a los resultados obtenidos, se destaca el clasificador MultinomialNB ya que entrega el mejor valor de AUC además de ser el más rápido en entregar resultados.

4.4.2. Clasificador de polaridad

El clasificador de polaridad es un problema de clasificación binaria cuyo objetivo es determinar si la respuesta del socio es positiva o negativa por medio de aprendizaje automático supervisado tomando como fuente el corpus anotado con la polaridad de la respuesta del socio.

Las distribuciones de los datos para este clasificador tampoco se encuentran muy balanceados ya que existe un 73,80% de respuestas que tienen una polaridad positiva, mientras que el 26,20% de respuestas tienen una polaridad negativa.

Para entender mejor el contiendo de cada grupo de la distribución se utiliza una nube de palabras, que permite tener una idea de los elementos de cada grupo.

Ilustración 19 Nube de palabras con polaridad positiva

En la clase con polaridad positiva se puede observar que las palabras que corresponden al saludo tienen una alta concentración, al igual que la palabra mañana que hace referencia a un posible compromiso de pago de acuerdo a las respuestas de los socios.

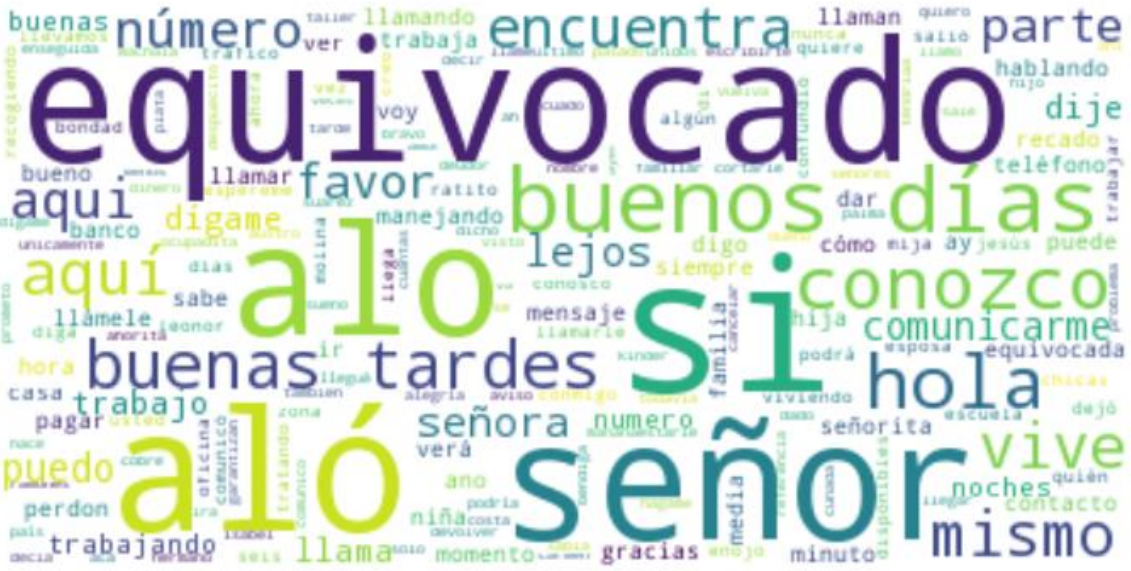


Ilustración 20 Nube de palabras de la polaridad negativa

Como se puede observar en la clase negativa, si bien existe una presencia relevante de las palabras que representan el saludo que da el socio al responder la llamada, existe una presencia mucho más representativa de la palabra equivocado, que indican que se contactaron con otra persona.

Se repite el procedimiento previamente descrito en el clasificador de relevancia, empezando la experimentación con el clasificador LinearSVC de SVM, con el cual se determinó las características más relevantes. Posteriormente se empieza a experimentar con el resto de algoritmos para obtener los resultados más significativos y establecer el mejor clasificador de acuerdo con una búsqueda exhaustiva con validación cruzada como métrica de evaluación AUC.

Se empieza el procedimiento con la selección del uso de vector de palabras con frecuencia de términos o no.

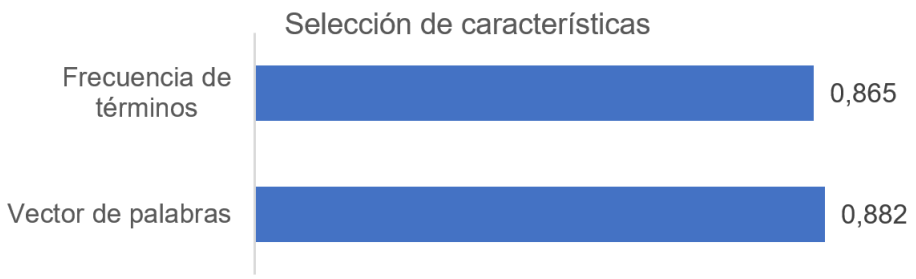


Ilustración 21 Vector de palabras vs TFM

Se puede observar que sigue prevaleciendo el uso de vector de palabras, sin la necesidad de utilizar la frecuencia de los términos. No se alcanza el mismo nivel de AUC que en el clasificador de relevancia, pero sigue existiendo una superioridad que se inclina hacia el vector de palabras.

En la experimentación de este clasificador se observa un comportamiento diferente con la combinación de términos, es decir en la generación de unigramas o bigramas, resultado de este comportamiento se presenta en la ilustración a continuación:

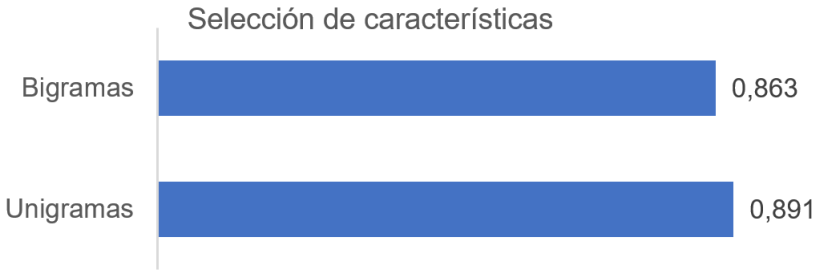


Ilustración 22 Unigramas vs Bigramas en clasificador de polaridad

En este clasificador prevalece el uso de unigramas para entregar los mejores resultados. A pesar de que los bigramas no generan una diferencia significativa, se utiliza el que entrega un mayor valor.

Para el resto de pruebas se repite el procedimiento. Para sintetizar estas pruebas, las características más representativas se han tomado por medio de búsqueda exhaustiva de dos conjuntos de parámetros.

Por medio de un pipeline se conectan los procesos en la secuencia de ejecución, empezando con el clasificador SVM (Scikit-learn, s.f.):

```
pipeline = Pipeline([
('vect', vectorizer),
('cls', LinearSVC()),
])
```

Se realizó una búsqueda exhaustiva por medio de GridSerchCV. Obviamente los valores de los parámetros dependen de cada clasificador y los posibles valores que pueden tomar dependen de cómo influyen estos valores en el dominio del problema, determinando el mejor juego de parámetros. Por ejemplo, para determinar los mejores parámetros del clasificador SVM se utiliza el siguiente pedazo de código (Scikit-learn, s.f.):

```
parameters = {
'vect__max_df': (0.5, 1.9),
'vect__min_df': (10, 20, 50),
'vect__max_features': (100, 500),
'vect__ngram_range': ((1, 1), (1, 2)),
'cls__C': (0.2, 0.5, 0.7),
'cls__loss': ('hinge', 'squared_hinge'),
'cls__max_iter': (500, 1000)
}
grid_search = GridSearchCV(pipeline, parameters, n_jobs=-1 ,
scoring='roc_auc')
```

A este conjunto de parámetros se lo llamará Test 1, con el que se realiza la búsqueda exhaustiva junto al estimador AUC para su comparación. Se reutilizan los conjuntos de parámetros con los otros clasificadores, excepto los que son propios del clasificador.

Se modifica el valor de los parámetros de Test 1 para obtener nuevos resultados con los cuales se comparará con los valores de AUC.

Continuando con el ejemplo, el conjunto de parámetros se modifica de la siguiente manera:

```
parameters = {
    'vect__max_df': (0.5, 1.0, 1.5, 2.0),
    'vect__min_df': (1, 5, 10, 20, 50),
    'vect__max_features': (100, 250, 500, 1000),
    'vect__ngram_range': ((1, 1), (1, 2)),
    'cls__C': (0.1, 0.5, 0.9),
    'cls__loss': ('hinge', 'squared_hinge'),
    'cls__max_iter': (500, 1000)
}
```

Este nuevo conjunto de parámetros se lo llamará Test 2. Con este nuevo conjunto se repite la búsqueda exhaustiva para cada uno de los clasificadores.

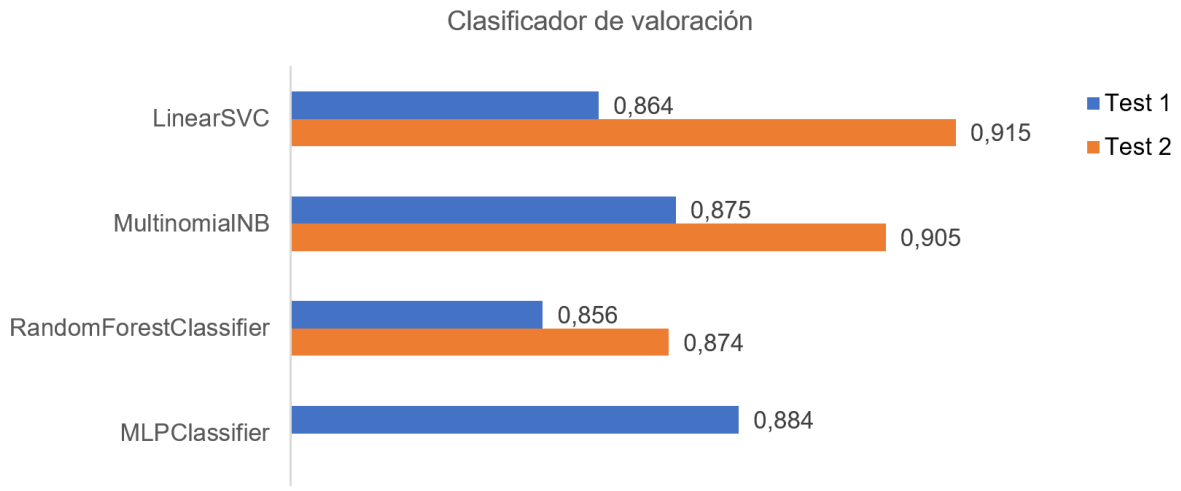


Ilustración 23 Comparación de resultados de los algoritmos de clasificación de polaridad

Se puede observar que las medias obtenidas de los clasificadores están todas por encima del 0.80 de AUC. El clasificador LinearSVC llega a obtener con el conjunto de parámetros de Test 2, el valor más alto de todos los clasificadores con una media de 0.915 de AUC.

El costo computacional ligado a estas iteraciones está ligado al tiempo en el que el clasificador llego a entregar los resultados en las diferentes combinaciones. El clasificador basado en una red neuronal para el conjunto de parámetros Test 2 no llegó a converger, lo cual se puede visualizar en la siguiente ilustración.

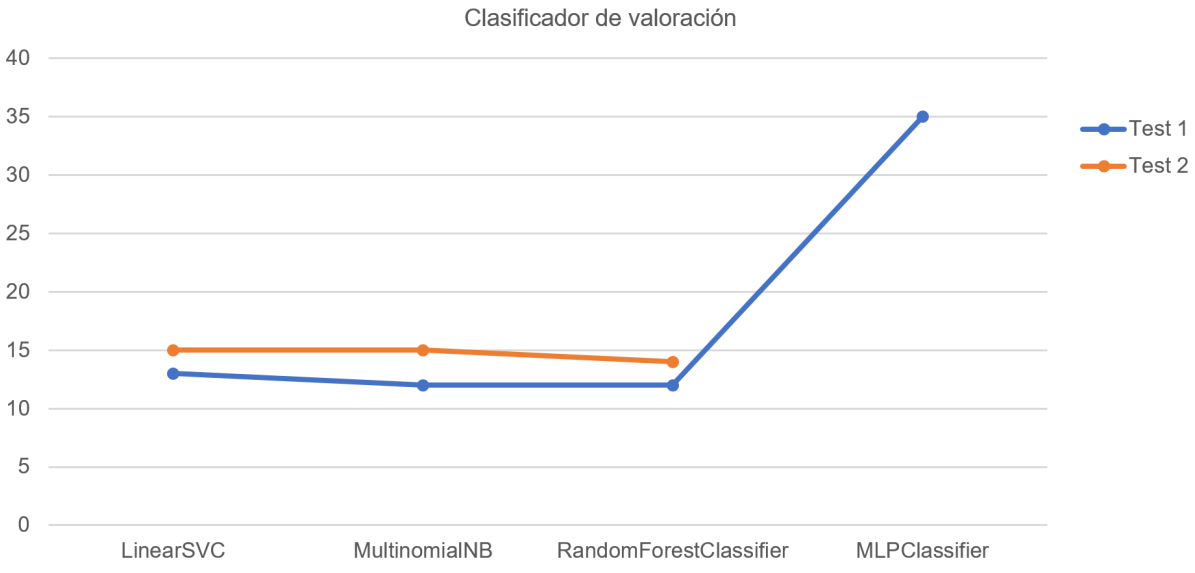


Ilustración 24 Tiempo de convergencia de los clasificadores de polaridad

En el modelo de clasificación de polaridad de acuerdo con los resultados obtenidos, se destaca el clasificador LinearSVC ya que entrega el mejor valor de AUC además de manejar un tiempo aceptable en entregar resultados en comparación con los otros clasificadores.

4.5. Evaluación de resultados

En la evaluación de los resultados se comprueba la precisión de los modelos definitivos con los conjuntos de parámetros de la etapa de experimentación, además si cumplen o no los objetivos planteados en el trabajo.

4.5.1. Clasificador de relevancia

Basándose en la ilustración 17 el mejor clasificador fue MultinomialNB. Una de las metas fue conseguir un clasificador que consiga un valor de AUC de 0.9, en realidad todos los clasificadores para este problema superaron este valor, aunque el valor más alto de 0.977 fue encontrado con el siguiente conjunto de parámetros:

Parámetro	Descripción	Valor
alpha	Valor de suavizado	0.001
max_df	Ignora los términos que tienen una frecuencia superior al valor	0.5
max_features	Número máximo de características	500
min_df	Ignora los términos que tienen una frecuencia inferior al valor	5
ngram_range	n-gramas a ser generados	(1, 2)

Tabla 9 Parámetros óptimos para el clasificador de relevancia

Con estos valores se realiza la evaluación de los datos de entrenamiento con validación cruzada para mitigar el sobreajuste, el resultado se evalúa por medio de la siguiente matriz de confusión:

Actual / Predicción	No relevante	Relevante
No relevante	7	6
Relevante	0	58

Tabla 10 Matriz de confusión del clasificador de relevancia

En donde podemos apreciar que las instancias correctamente clasificadas fueron 58 como relevantes y 7 como no relevantes, mientras que hubo 6 instancias que fueron incorrectamente clasificadas como relevantes y ninguna como no relevante.

A partir de la matriz de confusión se calculan las siguientes métricas:

Métrica	Descripción	Valor
Exactitud del clasificador	En general cuan correcto es el clasificador	0.915
Error del clasificador	En general cuan incorrecto es el clasificador	0.084
Sensibilidad	Predicciones correctas al clasificar instancias positivas	1
Especificidad	Predicciones correctas al clasificar instancias negativas	0.538
Tasa de falsos positivos	Predicciones incorrectas al clasificar instancias negativas	0.461
Precisión	Precisión del clasificador para predecir instancias positivas	0.906

Tabla 11 Métricas del clasificador de relevancia

Por medio de una representación gráfica de la curva ROC (Receiver Operating Characteristic o Características Operativa del Receptor) se visualiza la sensibilidad frente a la especificad para el clasificador, representada por la tasa de verdaderos positivos contra la tasa de falsos positivos.

La diagonal de la gráfica divide el espacio: los valores por encima de esta diagonal demuestran buenos resultados del clasificador, y los valores debajo de la diagonal son malos resultados.

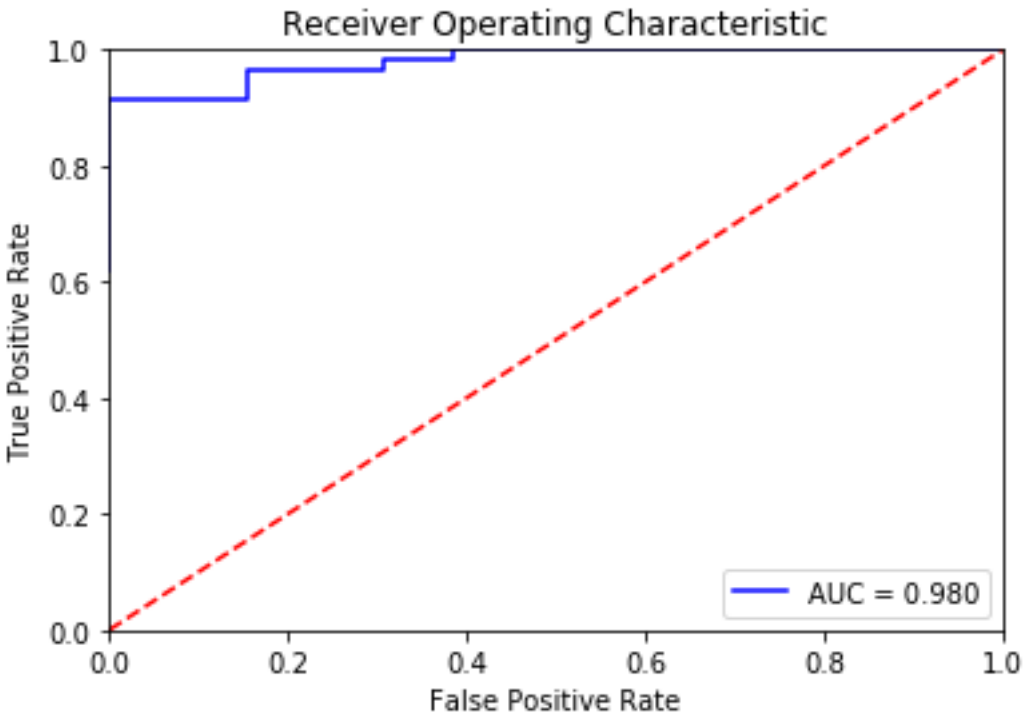


Ilustración 25 ROC AUC Clasificador de relevancia

El área bajo la curva se puede interpretar como la probabilidad de que el clasificador determine una instancia como relevante sobre una instancia no relevante. Como muestra la ilustración, un AUC de 0.98 indica que existe una probabilidad de 98% de que una instancia sea clasificada como relevante.

4.5.2. Clasificador de polaridad

Con referencia a la ilustración 22, el mejor clasificador fue LinearSVC. Con este clasificador se pretendía conseguir un valor de AUC de 0.9, pero ninguno de los algoritmos de clasificación pudo alcanzar este valor con el conjunto de parámetros Test 1. Con el conjunto de parámetros Test 2 los clasificadores MultinomialNB y LinearSVC, en el cual LinearSVC consiguió el valor más alto 0.915 para este problema, con el siguiente conjunto de parámetros:

Parámetro	Descripción	Valor
C	Parámetro de penalización del termino de error	0.1
loss	Función de perdida	squared_hinge
max_iter	Número máximo de iteraciones a ejecutar	500
max_df	Ignora los términos que tienen una frecuencia superior al valor	0.5
max_features	Número máximo de características	500
min_df	Ignora los términos que tienen una frecuencia inferior al valor	1
ngram_range	n-gramas a ser generados	(1, 1)

Tabla 12 Parámetros óptimos para el clasificador de polaridad

Con estos parámetros se realiza la evaluación de los datos de entrenamiento con validación cruzada para mitigar el sobreajuste, el resultado se evalúa por medio de la matriz de confusión que se muestra a continuación:

Actual / Predicción	Negativa	Positiva
Negativa	11	6
Positiva	2	40

Tabla 13 Matriz de confusión del clasificador de polaridad

Existen un total de 40 instancias que han sido correctamente clasificadas como positivas, 11 instancias correctamente clasificadas como negativas, 6 instancias que fueron incorrectamente clasificadas como positivas, 2 instancias clasificadas incorrectamente como negativas.

A partir de la matriz de confusión se calculan las siguientes métricas:

Métrica	Descripción	Valor
Exactitud del clasificador	En general cuan correcto es el clasificador	0.864
Error del clasificador	En general cuan incorrecto es el clasificador	0.135
Sensibilidad	Predicciones correctas al clasificar instancias positivas	0.952
Especificidad	Predicciones correctas al clasificar instancias negativas	0.647
Tasa de falsos positivos	Predicciones incorrectas al clasificar instancias negativas	0.352
Precisión	Precisión del clasificador para predecir instancias positivas	0.869

Tabla 14 Métricas del clasificador de polaridad

Por medio de una representación gráfica de la curva ROC se visualiza la tasa de verdaderos positivos contra la tasa de falsos positivos.

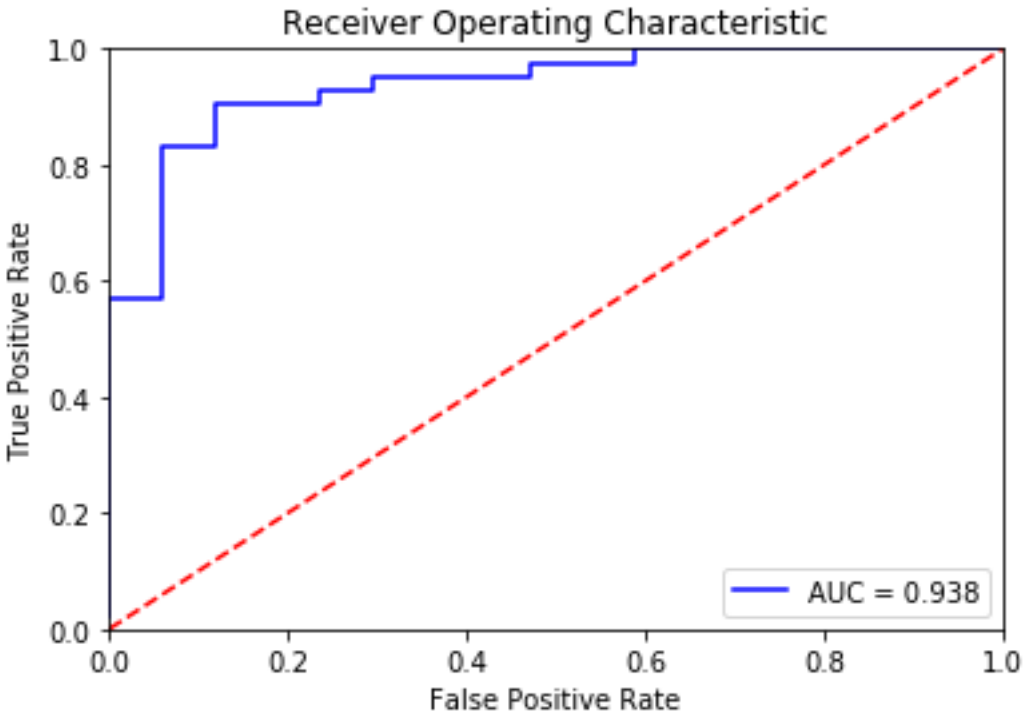


Ilustración 26 ROC AUC Clasificador de polaridad

El área bajo la curva AUC de 0.938 indica que existe una probabilidad de 93.8% de que una instancia sea clasificada como positiva.

4.6. Despliegue

En esta parte se describe el funcionamiento que el sistema tendrá en producción con los modelos definitivos. El sistema trabaja con la modalidad de procesamiento de lotes, no trabaja en línea.

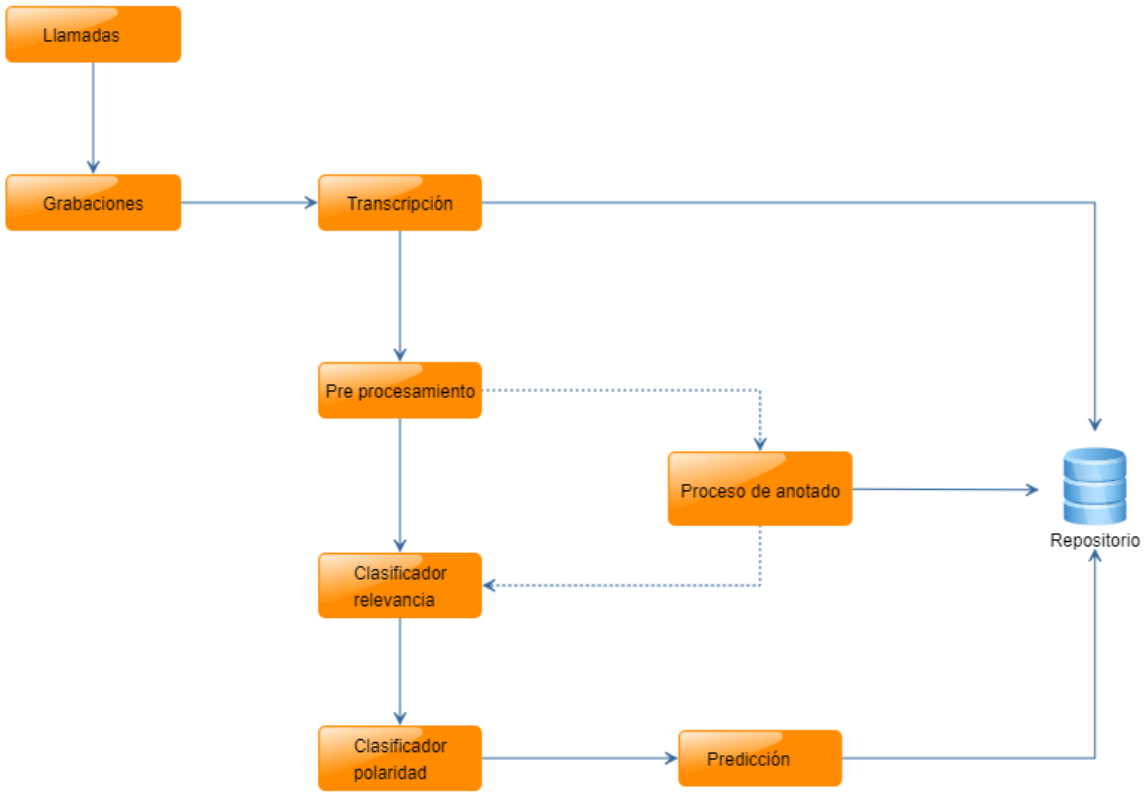


Ilustración 27 Procesamiento de clasificadores

El sistema se alimenta de las llamadas que realizaron los gestores para contactar a los socios, las conversaciones establecidas se almacenan estas como audios, a las cuales el sistema accede. Los procesos propios del sistema son los siguientes:

- Se transcriben los audios a textos, almacenándolos en un repositorio persistente que luego puede ser consultado.
- Se realiza una limpieza básica de los textos.
- El clasificador de relevancia determina si la respuesta del socio tiene información para ser valorada.

- El clasificador de polaridad determina si la respuesta del socio es negativa o positiva.
- El resultado de los clasificadores, la predicción, se estructura con un formato fácil de entender, mismo que es almacenado en un sistema persistente para consultas posteriores.

Se recomienda ejecutar el sistema por medio de tareas programas que procesen diariamente las grabaciones, durante el período de tiempo en el que no se puede contactar al socio por ser un horario inapropiado, es decir de las 21:00 en adelante. Este tiempo se podría utilizar en la tarea de clasificación diaria.

El sistema debe tener una etapa de mantenimiento, básicamente consiste en el reentrenamiento de los modelos, actividad que se recomienda realidad cada 3 meses, para lo cual se repetirán los procesos de este trabajo, actualizando los conjuntos de datos de entrenamiento y evaluación, ampliando el corpus inicial.

Además, se propone que el proceso de anotado manual se lo revise de ser posible con la misma frecuencia que el reentrenamiento, debido a que el proceso de transcripción no tiene un rendimiento adecuado con las grabaciones, lo cual puede afectar a los clasificadores en el resultado de nuevas instancias a categorizar.

Se puede tomar los datos de los últimos 3 meses anteriores a la fecha de reentrenamiento como el conjunto de datos de evaluación, el resto de datos se incluiría en el corpus entrenamiento.

5. Conclusiones y trabajo futuro

5.1 Conclusiones

En esta memoria de trabajo de fin de master se ha presentado una propuesta para realizar un proceso de calificación de la respuesta de un socio o cliente, desde el contexto de la gestión de cobranzas.

Para este problema existen soluciones ya desarrolladas en el campo de la minería de textos, que no pueden ser aplicados en primera instancia en este trabajo, ya que las respuestas se encuentran almacenadas en grabaciones de audio.

Por tanto, es necesario un procesamiento previo, utilizando el reconocimiento automático de voz, que ayude a transcribir la información contenida en un audio a texto, de forma que se puedan aplicar las soluciones de minería de textos

En este proceso se produjo un alto grado de error en las transcripciones de las grabaciones con todos los motores de reconocimiento utilizados, por lo que se recurrió a realizar una verificación manual de las transcripciones y correcciones en los casos necesarios.

Con la finalidad de mejorar el resultado de las transcripciones se debería profundizar en el entrenamiento de un modelo a medida que contemple las particularidades del idioma, modismo, acentos, entre otros, que contiene el audio de las grabaciones, con un motor como Kaldi ASR que tiene una buena comunidad detrás de este motor.

En el proceso de anotado manual se constató que las llamadas son generalmente cortas, debido a que en la mayoría de los casos no se contacta directamente al deudor, sino a personas que el deudor dio como referidos, a quienes se les pide que entreguen el mensaje del pago pendiente.

Los motivos por los cuales contactar con el deudor es complicado, son varios, entre los que se ha podido identificar:

- Condiciones laborales del deudor, tipo de trabajo, horarios, lugar de trabajo.
- Información desactualizada de los números de contacto del deudor.
- Evasión o bloqueo que realiza el deudor, debido a que tiene identificado el o los números de teléfono con los que se contactan los gestores de cobranzas.

Con la finalidad de calificar la respuesta del socio como positiva o negativa, se aplica un filtro previo que analiza si existe la suficiente cantidad de información, como un detector de

spam, que ha ayudado que los resultados del clasificador de polaridad tengan muy buenos resultados.

En el desarrollo del problema de clasificación de relevancia como se suponía en primera instancia, la mayoría de las repuestas de los socios fueron relevantes, lo cual se pudo verificar en el proceso de anotado manual. También se determinó que una respuesta es irrelevante porque el socio nunca contestó, o contestó la grabación automática indicando que deje el mensaje en el buzón de voz.

Se presentó un gran desequilibrio de las respuestas relevantes contra las no relevantes, que es un factor común en la clasificación de opiniones irrelevantes. Para poder superar este escenario se debería incrementar el tamaño de la muestra, y aplicar sobre la muestra original estratificación para llegar a balancear las instancias de la clase relevante y no relevante.

Sobre el conjunto de datos o corpus inicial, se han aplicado y comparado diversos métodos de aprendizaje automático supervisado. En cada uno de estos métodos se ha aplicado validación cruzada, con la finalidad de comparar los resultados y garantizar en la medida de lo posible una buena generalización en los modelos conseguidos, considerando el costo computacional de cada uno.

Los dos algoritmos con los cuales se empezó a comparar fueron SVM y Navies Bayes de acuerdo a lo citado en el estado de arte. Se han utilizado tradicionalmente en problemas de clasificación ya que a pesar de no ser modernos, han entregado buenos resultados en las pruebas iniciales.

Se utilizaron algoritmos basados en arboles de decisión, en donde se crea un grupo de clasificadores, que deberían ayudar a mejorar la predicción del objetivo. Así como también algoritmos basados en redes neuronales con múltiples capas con un costo computacional mucho más alto. El mejor clasificador para el primer nivel fue MultinomialMB mientras que pasa el segundo nivel LinearSVC.

Para determinar el espacio de características relevantes se realizó un experimento escogiendo un algoritmo sobre el cual se empezó a cambiar los parámetros y tomar los resultados para compararlos, en un proceso manual y repetitivo. Por medio de una búsqueda exhaustiva, que permite la herramienta utilizada, se encontró de una manera mucho más rápida los mejores parámetros para el modelo.

El lenguaje de programación en el cual se desarrolló el trabajo fue Python, muy utilizado dentro del mundo de la ciencia de datos y machine learning, utilizando scikit-learn como el

paquete de machine learning listo para usar, con una curva de aprendizaje corta, mediante Anaconda Plataform que es una distribución que incluye varios paquetes con las dependencias necesarias para usarlos inmediatamente.

Los procesos en los que se empleó más tiempo fueron la validación de la transcripción y el proceso de anotado manual de las transcripciones, en donde con un criterio de un experto humano se establecieron las guías para determinar que se considera relevante o no, así como que se considera como positivo o negativo.

Se ha alcanzado el valor de la métrica AUC para ambos clasificadores, que se establecieron en los objetivos, estos valores son razonables para el tamaño del conjunto de datos del corpus inicial. Los resultados obtenidos se pueden considerar buenos, aunque no se los ha podido comparar con otros trabajos debido a que no se encontró referencias similares más allá de la literatura indicada en el capítulo correspondiente al estado del arte.

El modelo de calificación actual puede ser generalizado, ya que los criterios de evaluación se establecen por un experto humano en el proceso de anotado manual. Por lo tanto, el resto de procedimientos pueden ser utilizados para adaptarse de una manera sencilla a otros modelos en los cuales se desea calificar respuestas u opiniones de socios, clientes o usuarios.

5.2 Líneas de trabajo futuro

Para continuar con el trabajo desarrollado, se pueden explorar las siguientes ideas:

- Desarrollar un modelo de reconocimiento de voz personalizado.
- Incrementar el tamaño del conjunto de datos para el entrenamiento, con la finalidad de aplicar otras técnicas de aprendizaje que se ajusten a tamaños más representativos.
- Agregar meta información como valoraciones realizadas por supervisores o los gestores al espacio de características.
- Utilizar enfoques más recientes como word2vec para la selección de características o el uso de deep learning, por ejemplo: Keras, para el modelado.
- Balancear la cantidad de instancias de las clases, relevante, no relevante, positiva y negativa.

- Utilizar enfoques de análisis semánticos, sintácticos y léxicos para complementar los modelos.

Desde un punto de vista más ambicioso, se podría considerar evaluar los sentimientos directamente en el habla, basándose en características acústicas, como el tono de voz, el volumen que se utiliza, entre otras.

6. Bibliografía

Bakliwal, A., Arora, P., Madhappan, S., Kapre, N., Singh, M., y Varma, V. (2012). Mining sentiments from tweets. *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis,* 11-18.

Bing Speech API. (s.f.). Speech Recognition Microsoft Azure. Recuperado de https://azure.microsoft.com/en-us/services/cognitive-services/speech/ [Consulta: 30 de mayo de 2017]

Blair-Goldensohn, S., Hannan, K., McDonald, R., Neylon, T., Reis, G. A., & Reynar, J. (2008). Building a sentiment summarizer for local service reviews. In *WWW workshop on NLP in the information explosion era* (Vol. 14, pp. 339-348).

Cortes, C., y Vapnik, V. (1995). Support-vector networks. *Machine Learning,* 20(3), 273-297. doi: 10.1007/BF00994018

Chowdhury, G. G. (2003). Natural language processing. *Annual review of information science and technology, 37*(1), 51-89.

CMUSphinx. (2017). Recuperado de https://en.wikipedia.org/wiki/CMU_ Sphinx [Consulta: el 30 de mayo del 2017]

Gaida, C., Lange, P., Petrick, R., Proba, P., Malatawy, A., & Suendermann-Oeft, D. (2014). Comparing open-source speech recognition toolkits. *Tech. Rep., DHBW Stuttgart.* Recuperado de http://suendermann.com/su/pdf/oasis2014.pdf [Consulta: 31 de mayo de 2017]

Hatzivassiloglou, V., & McKeown, K. R. (1997). Predicting the semantic orientation of adjectives. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics* (pp. 174-181). Association for Computational Linguistics.

Hiroshi, K., Tetsuya, N., & Hideo, W. (2004). Deeper sentiment analysis using machine translation technology. In *Proceedings of the 20th international conference on Computational Linguistics* (p. 494). Association for Computational Linguistics.

Israel, G. (1992). *Determining Sample Size. Fact Sheet PEOD-6. Program Evaluation and Organizational Development Series*. University of Florida, Florida. Recuperado de http://www.sut.ac.th/im/data/read6.pdf [Consulta: 15 de mayo de 2017]

Joachims, T. (1998). *Making large-scale SVM learning practical* (No. 1998, 28). Technical Report, SFB 475: Komplexitätsreduktion in Multivariaten Datenstrukturen, Universität Dortmund.

Juang, B.H., y Rabiner, L. (2005). *Automatic Speech Recognition - A Brief History of the Technology Development.* Santa Bárbara, California. Recuperado de: http://www.ece.ucsb.edu/Faculty/Rabiner/ece259/Reprints/354_LALI-ASRHistory-final-10-8.pdf [Consulta: 30 de Mayo del 2017]

Kaldi ASR. (s.f.). Recuperado de http://kaldi-asr.org/ [Consulta: 30 de mayo del 2017]

Kaldi-ASR (2017). Recuperado de https://github.com/kaldi-asr/kaldi [Consulta: 30 de mayo del 2017]

Keller, F. (2002). Naive Bayes Classifiers. *Connect. Stat. Lang. Process. Course Univ. Saarlandes.*

Këpuska, V., y Bohouta, G. (2017). Comparing Speech Recognition Systems. *Engineering Research and Application*, 7(3), 20-24.

Kononenko, I. (1990). Comparison of inductive and naive Bayesian learning approaches to automatic knowledge acquisition. *Current Trends in Knowledge Acquisition.* (194-197). Slovenia, IOS Press.

Liu, B. (2010). Sentiment Analysis and Subjectivity. *Handbook of natural language processing, 2*, 627-666.

Liu, B. (2012). *Sentiment Analysis and Opinion Mining* Recuperado de https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf [Consulta: 10 de junio de 2017]

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval, 2*(1–2), 1-135.

Pang, B., Lee, L., y Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. *Proceedings of the ACL-02 conference on Empirical methods in natural language processing, 10*, 79-86. doi: 10.3115/1118693.1118704

Nasukawa, T., & Yi, J. (2003). Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture* (pp. 70-77). ACM.

Natural Language Toolkit (s.f.). NLTK 3.2.4 documentation. Recuperado de http://www.nltk.org [Consulta: 15 de junio de 2017]

Rabiner, L. R., & Juang, B. H. (1993). *Fundamentals of speech recognition*. Prentice Hall.

Scikit-learn. (s.f.). Recuperado de http://scikit-learn.org/stable/ [Consulta: 15 de junio del 2017]

Sokolova, M. y Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Journal Information Processing and Management: An International Journal, 45*(4), 427-437. doi: 10.1016/j.ipm.2009.03.002

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR), 34*(1), 1-47.

Shmyrev, N. (s.f.). CMUSphinx Open Source Speech Recognition. Recuperado de https://cmusphinx.github.io/ [Consulta: 30 de mayo del 2017]

Speech API - Speech Recognition (s.f.). Google Cloud Platform. Recuperado de https://cloud.google.com/speech/ [Consulta: 30 de mayo del 2017]

Speech recognition (2017). Recuperado de https://en.wikipedia.org/wiki/Speech_recognition [Consulta: 30 de mayo del 2017]

Suárez, E. J. C. (2014). Tutorial sobre máquinas de vectores soporte (sVM). *Tutorial sobre Máquinas de Vectores Soporte (SVM).* [Consulta: 15 de junio de 2017] http://www.ia.uned.es/~ejcarmona/publicaciones/[2016-%20Carmona]%20SVM.pdf

Turney, P. (2002). Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews*. Proceedings of the 40th Annual meeting of the Association for Computational Linguistics*, 417-424. doi: 10.3115/1073083.1073153

Vapnik, V. N., & Vapnik, V. (1998). Statistical learning theory (Vol. 1). New York: Wiley.

Wiebe, J., Bruce, R., y O'Hara, T. (1999). Development and use of a gold standard dataset for subjectivity classifications. *Proceedings of the 37th annual meeting of the Association for Computational Linguistics,* 246-253. doi: 10.3115/1034678.1034721

Yi, J., Nasukawa, T., Bunescu, R., & Niblack, W. (2003). Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on* (pp. 427-434). IEEE.