Detección de contenido web asociado a temas de Lavado de Activos y Financiación del Terrorismo

Felipe Jiménez Leandro

Resumen

Este documento detalla la creación de un prototipo que estima la probabilidad de que una noticia de la *web* tenga contenido de temática de lavado de activos, financiación del terrorismo o temas conexos (en todo el mundo y en diversos de idiomas), de tal manera que se puedan detectar entidades vinculadas a un grupo financiero (clientes, gestores, proveedores) que tengan contenido relacionadas a esos temas, para poder realizar la debida diligencia de dichas entidades y determinar que acción procede basada en la severidad del caso. Mediante la creación de arañas *web*, para la extracción de información automática, el procesamiento del lenguaje natural para obtener la información relevante de los textos y un procedimiento de consenso de diversos algoritmos de aprendizaje estadístico para generar los modelos que asignan la probabilidad y clasifican los textos, se logra de manera exitosa el objetivo con una alta precisión (Curva ROC 98% en la muestra de validación) en los modelos de clasificación y una alta discriminación en la prueba de concepto realizada, utilizando individuos ya conocidos por sus actividades confirmadas en temas de lavado de dinero. Se detalla en el documento las aplicaciones que se podrían derivar sobre estas ideas.

Introducción

¿Las organizaciones financieras se encuentran haciendo uso hoy día de la *Big Data* almacenada en la nube y que es de acceso público?

Con la gran cantidad de información producida diariamente en el mundo, muchas de las empresas líderes e innovadoras en sus ámbitos han nacido en el paradigma *data-driven* o se han convertido a este. Compañías como Google, Amazon, Netflix, Facebook, Twitter, Spotify entre muchas otras, basan sus negocios en los datos. En los grandes datos. Para esto han desarrollado muchas tecnologías sacando provecho de la data producida por ellos mismos a cada instante, en volúmenes masivos y en gran cantidad de formatos de naturaleza no estructurada y por lo tanto de mayor dificultad para la extracción de información a partir de estos (McAfee & Brynjolfsson, 2012).

La última frase del párrafo anterior lleva a una nueva pregunta, ¿Cuánto es el provecho que las organizaciones obtienen de la data no estructurada?

La mayoría de información producida actualmente nace en formatos no estructurados o no tradicionales (textos, audios, videos, SMS, fotografías o imágenes, correos, *likes*, *twits*, etc), sin embargo, en los análisis sigue primando el uso de data estructurada para la toma de decisiones (Atre & Blumberg, 2003).

Este trabajo busca aumentar el conocimiento en el tema de la extracción de *data no estructurada* desde la nube, el tratamiento que se le debe hacer a esta *data* para poder reducirla y extraer solo su porción informativa y finalmente generar

modelos de *statistical learning* que logren realizar inferencias sobre nueva data. Lo anterior puede funcionar en gran una diversidad áreas para realizar aplicaciones que resuelvan problemas en el ámbito de los negocios.

En este caso, este trabajo nace debido a la oportunidad de utilizar este tipo de data en el ámbito del *Anti Money Laundering*, la cual aún no es explotada de forma sistemática. Algunos departamentos de Cumplimiento, que por lo general se encargan de esta tarea en las organizaciones de intermediación financiera, recolectan información de noticias de sus ámbitos geográficos de manera manual y poco automatizada, a costa del tiempo de sus analistas que podrían encontrarse en otras actividades más analíticas o investigativas, más allá de tareas operativas como la revisión de periódicos en busca de noticias de LAFT (Lavado de Activos/ Financiación de Terrorismo).

Otras cuentan con herramientas automáticas, pero por lo general estas se basan en *querys* tradicionales y no cuentan con modelos de aprendizaje automático.

El objetivo de este trabajo es generar un prototipo que busque de manera automática noticias en todo el mundo relacionadas con temas LAFT y que clasifique las noticias en la categoría de LAFT o en otro tema según una probabilidad estimada, por medio de técnicas analíticas, para una diversidad de fines que pueden funcionar como controles o monitoreos que coadyuden a los sistemas que generan alertas de posibles casos de actividad sospechosa de clientes pero que solo toman en cuenta información interna de la organización.

Algunos de los usos podrían ser los siguientes:

- Generación de alertas: Correr procesos automatizados sobre la cartera de clientes, gestores u ordenantes de operaciones, relacionados con el grupo, para medir la probabilidad de que los vinculados tengan indicios de temas LAFT en las noticias a nivel mundial.
- Consultas o investigaciones especiales a ciertos individuos, generadas generalmente en los departamentos de cumplimiento, por medio de una interfaz web gráfica.
- Monitoreo automatizado de noticias: Monitorear de manera automatizada las noticias por país o zona geográfica y las tendencias a través del tiempo, para ahorrar la tarea de búsqueda de noticias relacionadas al tema LAFT.
- Generación de métricas y scores para cuantificar el riesgo y las tipologías de LAFT por área geográfica y a través del tiempo.

Este documento detalla la metodología y técnicas con la que se puede ejecutar esta idea y los resultados obtenidos de la aplicación con los datos recopilados para probarla, además de pruebas de concepto para medir comprobar la efectividad de la aplicación con datos reales.

Metodología y datos

Toda la aplicación se desarrolló con el ambiente de computación estadística R. Se hizo uso de diversos paquetes que amplían la funcionalidad del ambiente a ámbitos más específicos que facilitan muchas de las tareas.

Para desarrollar la aplicación, se hace uso de tres diferentes tecnologías (*Web Crawling, Text Analytics* y *statistical learning)* que se utilizaron en etapas separadas del proyecto y que se citan a continuación:

1-Web Crawling

Las arañas web funcionan como extractoras de data proveniente de la Web de manera automatizada. En este caso la araña se programó para extraer noticias del agregador *Google News*, que contiene muchas de las fuentes más importantes de noticias del globo y que fue la fuente de datos de este trabajo. Se utilizó el paquete tm.plugin.webmining (Annau, 2015) para la extracción de las noticias que fueron automatizadas durante un periodo de tiempo.

Para poder extraer información relevante y estudiar las características que tienen las noticias LAFT en términos de contenido, se generó un diccionario con *tokens* (palabras o frases) referentes al tema LAFT y fraude. El diccionario es el siguiente:

***Tokens* en Español**

Lavado de activos, terrorismo, mafia, mariguana, contrabando, fraude, corrupción, cartel, cocaína, evasión de impuestos, extorsión, droga, malversación fondos,

crimen organizado, lavado dinero, trata personas, pitufeo, blanqueo capitales, narcotráfico, explotación sexual, decomiso, capo, fraude fiscal.

***Tokens* en Inglés**

Money Laundering, terrorism, marihuana, fraud, corruption, drug, tax evasion, extorsion, smuggling, cocaine, bitcoin, Human Trafficking, drug trafficking, hawala.

Con el anterior diccionario definido, se realizó una extracción automatizada diaria entre los meses de abril y junio del año 2015 de los textos o noticias que contuvieran los *tokens* del diccionario.

No obstante, para poder discriminar o distinguir cuándo un texto hace alusión a un tema LAFT o no, es importante contar con datos de textos que en efecto no son de temas LAFT. Para este propósito se utilizó en el mismo periodo una extracción automática de textos tomando como base *Stop Words* (palabras sumamente comunes en cada idioma que no tienen valor analítico) al azar.

Los resultados se guardan en objetos llamados *Corpus* que son archivos que contienen cada uno de los textos en crudo, pero además contienen metadata con información de cada texto recuperado, como el link de la página origen, la fecha de creación del texto, el autor, el idioma (en caso de ser guardado) entre otros.

2-Text Analytics

Los textos recuperados contienen mucha información poco valiosa mezclada con información valiosa. Este segundo grupo de herramientas brinda la posibilidad de

separar y extraer solamente la información valiosa y eliminar lo que no es necesario para el objetivo de este trabajo. A este paso se le llama limpieza de los datos. Además, estos datos se encuentran en forma no estructurada (data que no tiene un modelo de datos asociado y no tiene un formato predefinido) y debe ser llevados a formatos estructurados para ser analizados de una manera más sencilla. A este segundo punto se le llama transformación en datos estructurados. Este paso fue realizado mediante el paquete tm , que contiene funciones especiales para trabajar datos no estructurados en formato de texto.

2.1-Limpieza de los datos

A continuación se detalla cada uno de los pasos seguidos para realizar esta reducción de los datos y aislar solamente lo importante para el contexto de este prototipo (Feinerer, Hornik, & Meyer, 2008).

2.1.1- Remover símbolos

Se remueven todos los símbolos, puntuaciones y cualquier elemento extraño del *corpus*. Se debe tener cuidado de dejar espacio al removerlos para no formar nuevas palabras inexistentes.

2.1.2- Conversión a minúscula

Para reducir la cantidad de *tokens* que se utilizarán en los modelos, se pasan todos los elementos a minúscula, para que no se tomen en cuenta como elementos diferentes.

2.1.3- Remover *Stop Words*

Los *Stop Word* son palabras específicas de cada idioma que por su naturaleza son muy frecuentes pero poco útiles en materia analítica. Ejemplos de S*top Words* en el idioma español son: tendremos, ese, estad, estando, están, tenemos, este, estuviesen. Para este trabajo se removieron este tipo de palabras en español y en inglés, además se construyó un diccionario adicional de este tipo de palabras, basados en métricas que se definirán más adelante.

2.1.4- Remover espacio vacío extra

Se remueve espacio vacío extra para no afectar los *tokens* formados.

2.1.5- Remover números

Se remueven los números ya que para esta aplicación no son elementos a tomar a en cuenta debido a que no aportan valor analítico.

2.1.6- Remover acento

Debido a que las palabras que deben tener tildes no en todas ocasiones aparecen con tilde, se decide eliminar todas las tildes de ese tipo de palabras para que sean tomadas en cuenta como un solo elemento. Esto se puede generalizar para otros tipos de acentos.

2.1.7- Remover textos asociados a temas LAFT

Para la categoría de textos No LAFT, se deben eliminar los textos que sean LAFT, ya que estos fueron recuperados al azar y por lo tanto podrían haberse incluido

algunas noticias LAFT en esta categoría que podría dificultar la discriminación en los modelos. Para esta aplicación se eliminaron todos los textos que tuvieran presencia de alguna de las palabras del diccionario LAFT construido.

2.1.8- Remover textos en otros idiomas

A pesar de que la extracción se especificó solo diccionarios en español o en inglés, se recuperaron textos en otros idiomas debido a la similaridad de ciertas palabras. Se tomó la decisión de incorporar a los modelos solo textos en español o en inglés por lo que se removieron todos los idiomas adicionales. Para esto se utilizó una función que compara el conjunto de palabras de cada texto con el perfil predefinido de varios idiomas y asigna el de mayor similitud.

2.1.9- Aplicar sinónimos

Ciertos *tokens* significan lo mismo para el contexto de esta aplicación o cualquier tipo de aplicación. Por esto es conveniente reducir la cantidad de elementos que se incorporarán en los modelos para disminuir la variabilidad y robustecer los patrones de cada *token*. Por ejemplo, "lavado de activos" es considerado un sinónimo de "lavado de dinero" o "blanqueo de capitales".

2.1.10- Aplicar *Stemming*

El *Stemming* es una técnica que aplica algoritmos para encontrar la raíz de las palabras (Lang, 2004). Esto es conveniente en el mismo sentido del punto anterior: reducir la cantidad de elementos a tomar en cuenta en el modelado,

sustituyendo distintas palabras por conceptos. Un ejemplo del resultado de la aplicación de *Stemming* es el siguiente:

Texto original

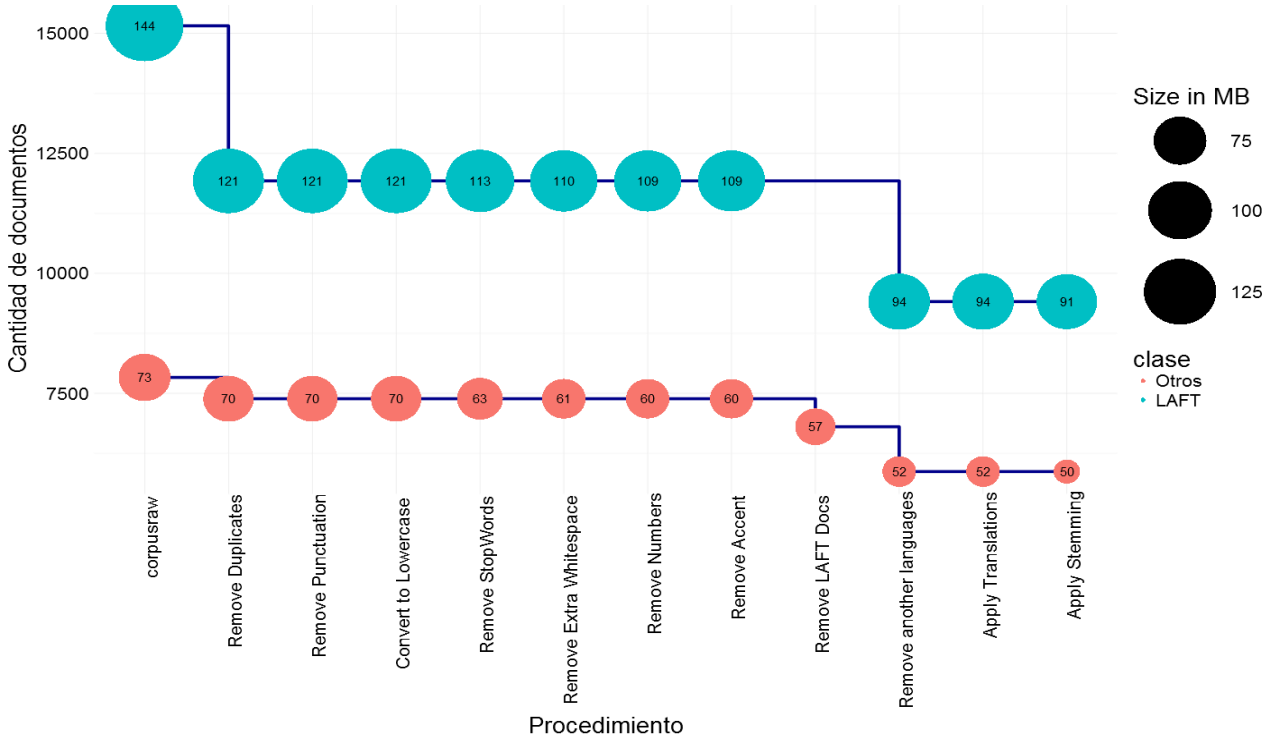
“Incluso los nombres de los cuatro ministerios que los gobiernan revelan un gran descaro al tergiversar deliberadamente los hechos. El Ministerio de la Paz se ocupa de la guerra; el Ministerio de la Verdad, de las mentiras; el Ministerio del Amor, de la tortura, y el Ministerio de la Abundancia, del hambre.” (Orwell, 1949)

Texto con aplicación de *Stemming*

*“*Inclus los nombr de los cuatr ministeri que los gobiern revel un gran descar al tergivers deliber los hechos. El Ministeri de la Paz se ocup de la guerra; el Ministeri de la Verdad, de las mentiras; el Ministeri del Amor, de la tortura, y el Ministeri de la Abundancia, del hambre.”

El gráfico 1 muestra la evolución en el tamaño del *corpus* al aplicar cada uno de los puntos revisados anteriormente:

Gráfico 1: Reducción en el tamaño de los corpus y cantidad de documentos



2.2- Transformación en datos estructurados

Una vez depurado y aplicada la limpieza al *Corpus* de todos los elementos sin valor analítico, se debe proceder a modificar la estructura de los datos para que puedan ser utilizados de una manera más conveniente por los modelos de aprendizaje automático. La secuencia de pasos en esta actividad se puede resumir en lo siguiente.

2.2.1- Construcción de *N-Grams*

Los *N-Grams* se utilizan para combinar dos o más palabras que por su alto grado de aparición conjunta conviene utilizarlos en los modelos para ganar información ya que se pueden referir a conceptos importantes (Buchta, y otros, 2013).

Por ejemplo "blanqueo capitales", "malversación de fondos" o "fraude fiscal" son ejemplos de *Bi-Grams*. En esta aplicación se utilizarán los *Bi-Grams*, ya que la mayoría de conceptos en temas LAFT no sobrepasan las dos palabras.

2.2.2.- Construcción de *Document Term Matrix (DTM)*

En este paso se transforma la data no estructurada del *Corpus* en una matriz en donde las filas son cada uno de los documentos o noticias y en las columnas se encuentran las variables (palabras o *bi-grams* que aparecen en las noticias). Se eliminan ciertas palabras que no cumplen con algunos parámetros definidos como el tamaño de la palabra (mínimo=4, máximo=infinito) y la cantidad de documentos en los que aparecen (mínimo=5, máximo=infinito).

2.2.3- Generar métrica Tf-Idf (*Term frecuency-Inverse document frecuency*)

La matriz de documentos y *tokens* debe ser llenada por alguna métrica que pondere cada palabra en función de su importancia en un documento en el *Corpus*. Se utiliza la métrica **Tf-Idf** ya que considera dos elementos importantes (Ramos, 2003) que se detallan a continuación.

Tf (Term frecuency)

Mide la frecuencia con la que el *token* **t** aparece en el documento **d**. Este indicador muestra que tan común es la aparición de cada término en cada uno de los documentos. Existen varias maneras de medirlo, entre ellas están las que se muestran en el Cuadro 1:

Cuadro 1: Métrica Tf

Esquema	Métrica Tf
Frecuencia Bruta	Cantidad de veces que aparece el *token* **t** en el documento **d**. Denotada como **f(t,d)**
Frecuencias Booleanas	1 si **t** ocurre en **d** y 0 si no ocurre
Frecuencias escalada logarítmicamente	1 + log f(t,d) y 0 si f(t,d)=0

Idf (Inverse document frecuency)

Este segundo componente pondera alto a los términos que no son muy comunes a través de todos los documentos, previniendo de no ponderar con valores altos las palabras comunes sin valor analítico. Las variantes para operacionalizar esta métrica se muestran en el Cuadro 2:

Cuadro 2: Métrica Idf

Esquema	Métrica Idf
Frecuencia inversa	$\mathrm{Log}\frac{\mathrm{N}}{\mathrm{n_t}}$
Frecuencia inversa suavizada	$\mathrm{Log}(1+\frac{\mathrm{N}}{\mathrm{n_t}})$
Frecuencia inversa probabilística	$\frac{\mathrm{N}-\mathrm{n_t}}{\mathrm{n_t}}$

Para el caso de esta aplicación se utiliza la siguiente combinación de **Tf-Idf:**

$$\mathbf{W_{t,f} = Log(1 + tf_{t,d}) * Log(\frac{N}{n_t})}$$

Por su naturaleza, la mayoría de matrices producto de esta transformación son matrices "escasas"(*sparse matrix*) en donde la mayoría de las casillas son ceros. Por este motivo se aplica una función para reducir la cantidad de términos que contribuyen a esta escasez de información. El parámetro definido fue de 0.98. Lo anterior quiere decir que todos los términos en los que el 98% o más de sus casillas sean ceros son removidos.

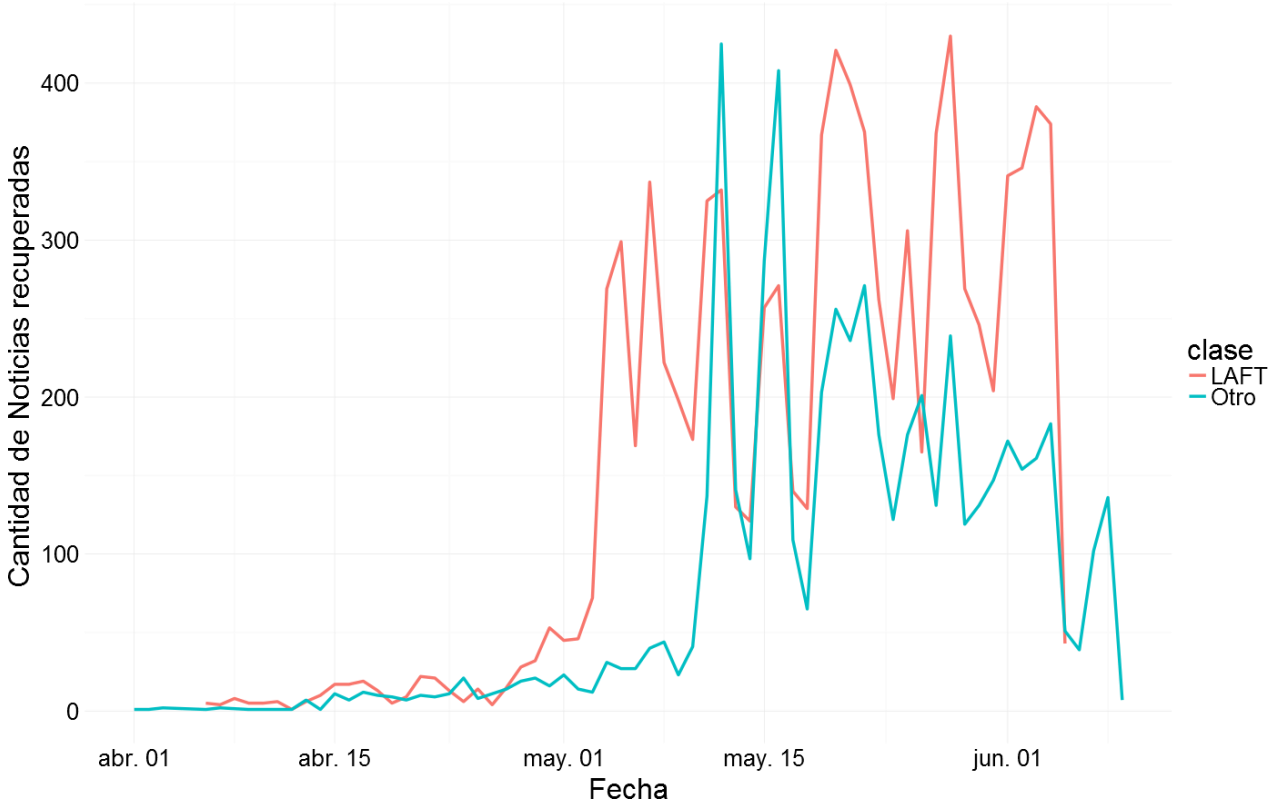
Un resumen de las matrices obtenidas es el siguiente:

Cuadro 3: Resumen del DTM de Categoría LAFT

Resultado	Categoría LAFT	Categoría General	Total
Documentos	9412	5870	15282
Términos	1485	1591	3076
% Casillas vacías	95%	95%	95%

El gráfico 2 muestra un resumen por día de la cantidad de noticias recuperadas para cada una de las categorías:

Gráfico 2: Cantidad de noticias recuperadas por fecha y tema



Se extrajo un total de 14944 documentos (9412 de temas LAFT y 5870 de otros temas) desde los meses de abril a junio del año 2015 (luego de aplicar filtros de remover duplicados).

Para verificar que los textos recuperados eran de temas LAFT, se generó un muestreo estadístico para cuantificar y verificar manualmente la proporción de textos que en efecto eran de temática LAFT. La proporción meta fue del 90%, no obstante el porcentaje en la muestra fue aún mayor (95%).

3-Statistical Learning:

El último eslabón metodológico es el que finalmente construye los modelos que clasificarán los textos en categoría LAFT u otros, con una probabilidad asociada a cada clase. Esta sección está basada en el libro "*Elements of Statistical Learning"* (Friedman, Hastie, & Tibshirani, Elements of Statistical Learning, 2009). El aprendizaje estadístico se refiere a un vasto juego de herramientas cuyo objetivo es entender los datos. Estas herramientas pueden ser clasificadas como supervisadas o no supervisadas. El aprendizaje supervisado, el cual es el utilizado en este trabajo, construye un modelo estadístico para predecir o estimar un resultado basado en una serie de características de entrada.

De esta manera, con los modelos supervisados de aprendizaje estadístico, se puede clasificar cada texto o noticia en la categoría LAFT o en la categoría de otros temas, en función del grado de parecido que cada noticia en específico tenga con la estructura de los textos ya clasificados en la muestra o *Corpus* recuperado. Por este motivo, textos con una alta densidad de *tokens*, palabras y temática de lavado de activos y financiación del terrorismo tendrán una alta probabilidad de ser clasificados por el modelo en esta categoría.

En el aprendizaje no supervisado no se genera una predicción o estimación, pero se obtiene información importante sobre la estructura y relaciones de los datos, utilizado en aplicaciones de segmentación o sistemas de recomendación.

Para todo el procedimiento se utilizó el paquete caret, que contiene una amplia gama de herramientas para entrenar y validar modelos de aprendizaje supervisados.

3.1-Algoritmos utilizados

En el panorama actual existe una diversidad de algoritmos de aprendizaje supervisado, tanto novedosos como clásicos cuyo grado de efectividad puede variar en función de qué tan adaptables sean al tipo de datos que se están utilizando. Por este motivo para construir este prototipo se escogen cuatro algoritmos de diversas corrientes con el objetivo de comparar su rendimiento en los datos recolectados y generar un modelo "consenso" que pondere o de más peso a los algoritmos que se hayan desempeñado mejor en las etapas de entrenamiento y prueba. Para todo este procedimiento y para todos los algoritmos A continuación se describen los algoritmos utilizados.

3.1.1-Random Forest (Bosques aleatorios)

Para introducir los bosques aleatorios, se deben explicar dos técnicas importantes:

Métodos de árboles: Los métodos de árboles son técnicas que sirven para estimar un valor de una variable (regresión) o asignar una categoría (clasificación) basado en las características de otras variables. Para esto se divide el espacio predictor en segmentos o estratos. Son métodos sencillos de interpretación y a su vez útiles. No obstante, estos métodos no son tan poderosos como los bosques aleatorios.

Bagging (Bootstrap aggregating): Este algoritmo fue diseñado para mejorar la estabilidad y precisión de los modelos de aprendizaje estadístico. Dado un juego de datos de entrenamiento **D** de tamaño **n**, se generan **m** nuevos juegos **Di** de tamaño **ni** muestreando con reemplazo de **D**. Las **m** muestras son ajustadas según el algoritmo utilizado y el resultado de estimación o predicción para cada observación es promediado (para variables continuas) o mediante un esquema de votación (para variables cualitativas).

Esta técnica tiende a mejorar los algoritmos inestables como las redes neuronales artificiales o los árboles de regresión y clasificación. No obstante, puede degradar el resultado de métodos estables como los *K-nearest neighbors* (Breiman, 1996)

Los bosques aleatorios, como el nombre lo sugiere, se trata de la aplicación de numerosos árboles para mejorar los resultados de precisión. Inclusive, los bosques tienen una ventaja sobre los métodos que aplican solo *bagging.* Como en el *bagging,* también se construyen **A** árboles por usando muestreo *bootstrap*. La diferencia radica en que cuando se construyen estos árboles en cada ejecución se considera un cambio en el número **m** de predictores escogidos de manera aleatoria. Al considerar diferentes predictores en cada árbol, la variabilidad obtenida será menor a un método de *bagging* y consecuentemente se lograrán árboles menos correlacionados y más confiables, debido a que se reduce el sobreajuste.

3.1.2-Gradient Boosting Models (Modelos de Potenciación del Gradiente)

Otro concepto importante en el aprendizaje estadístico es la técnica denominada *boosting*, que es un procedimiento para reducir el sesgo y la variancia de los modelos. Como el *bagging*, se trata de un enfoque general que puede ser aplicado a muchos métodos de aprendizaje estadístico tanto para problemas de clasificación como de regresión. La diferencia con el bagging radica en que los árboles son construidos secuencialmente. Es decir, cada árbol toma información del árbol anterior para disminuir el error en las áreas en donde el rendimiento no es el mejor. En el caso del problema de clasificación, se les da un mayor peso a los individuos mal clasificados, potenciándolos para que tengan mayor probabilidad de aparecer en el siguiente árbol y pueda ser mejorada su estimación. Básicamente, cada nuevo modelo es ajustado sobre los errores o residuales del modelo anterior.

Otra característica común en el *boosting* es que se los modelos utilizados son llamados "modelos débiles" con bajo poder predictivo, pero que al combinarlos generan un estimador con un alto nivel de predicción o precisión.

Al estar ajustando nuevos modelos con los residuales del modelo anterior, en realidad se está ajustando el gradiente negativo o lo que es conocido como el descenso del gradiente. El motivo por el cual se ajustan el descenso del gradiente y no los residuales, es debido a que este procedimiento es afectado en menor medida por los *outliers* (Friedman, Greedy function approximation: a gradient boosting machine, 2001).

El método es ajustado en base a tres parámetros:

El número de árboles B: El método puede tener sobreajuste si el número **B** es grande.

Parámetro de contracción lambda L: Un valor positivo pequeño. Controla la tasa a la que el método aprende. Valores pequeños necesitan de un **B** grande para obtener buenos resultados

El número d de cortes en cada árbol: Controla la complejidad del procedimiento general. Por lo general un d=1 trabaja bien, en este caso cada árbol tiene un solo corte y dos nodos terminales. En general, este parámetro mide la profundidad de las interacciones entre las variables utilizadas.

3.1.3-LogitBoost (Boosted Logistic Regression)

Este algoritmo también utiliza como base el *boosting*. Se escoge este algoritmo sobre el tradicional AdaBoost, debido a que usualmente trabaja mejor con data "ruidosa" como la que se utiliza en esta aplicación. En este caso solo se utilizan árboles con un solo corte (d=1), que solamente tienen dos nodos terminales y que cumplen con la característica de "clasificadores débiles". Utiliza una función de log-verosimilitud binomial para ajustar el descenso del gradiente (Buhlmann & Dettling, 2002).

3.1.4-Support Vector Machines (Máquinas Vectoriales de Soporte)

Las máquinas vectoriales de soporte (SVM por sus siglas en inglés) fueron desarrolladas en los años 90 en las escuelas de Ciencias de la Computación y han sido considerados como uno de los mejores clasificadores debido a su buen rendimiento en una variedad de aplicaciones.

Al igual que otras técnicas revisadas, el algoritmo construye un modelo que asigna las observaciones en una categoría, pero haciendo un clasificador no probabilístico, lineal y binario. El modelo es una representación de los puntos en un espacio, en el cual se traza una frontera (hiperplano) que divide las dos clases en el máximo margen posible. A los puntos que conforman las dos líneas paralelas a este hiperplano se les llama vectores de soporte, siendo la distancia entre ellas, el margen máximo posible.

Adicionalmente a la clasificación lineal, este algoritmo puede de una manera eficiente realizar modelado no lineal, basado en el "truco del kernel" con el cual se puede construir hiperplano de dimensionalidad muy alta o incluso infinita.

Para esta aplicación se utilizaron dos tipos de kernel: el "radial" y el "Polinomial".

3.2-Validación cruzada

Para garantizar la generalización de resultados, en muestras externas con características diferentes a las que se utilizan en el modelado y entrenar de una manera confiable los modelos para reducir su error de clasificación, se utiliza la técnica de validación cruzada.

Es necesario introducir dos conceptos utilizados en el aprendizaje estadístico:

Entrenamiento: Se refiere a la porción del juego de datos que será utilizado para que los modelos "aprendan" la estructura y relaciones en los datos para poder hacer las estimaciones. Por lo general se utiliza un porcentaje alto de la muestra en esta etapa.

Prueba: El porcentaje restante de datos permanece fuera del aprendizaje. Estos datos son utilizados para probar los modelos y medir el error que estos arrojan.

Lo anterior se realiza de esta forma ya que si se mide el error en la propia muestra de entrenamiento se tiende a subestimar el error debido a que los modelos aprenden a clasificar bien solo en la muestra pero no en datos externos que pueden tener características distintas.

La validación cruzada utilizada en esta aplicación es del tipo *k-fold* que consiste en dividir aleatoriamente el juego de datos en **K** particiones de un tamaño aproximado. La primera partición es usada como muestra de prueba y las restantes **k-1** son usadas para ajustar el modelo. De esta manera se computa el error de la primera partición. Posteriormente la segunda partición se utiliza como muestra de prueba y las restantes como muestra de entrenamiento, con el cálculo correspondiente del error. Esto se realiza hasta que se haya recorrido todas las particiones.

Con la siguiente fórmula se calcula el error de todo el procedimiento (Kohavi, 1995):

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

El procedimiento de validación cruzada puede realizarse solo una vez para medir el error de clasificación o puede realizarse repetidas veces para obtener una mayor confianza en el valor del error obtenido, en claro detrimento del tiempo computacional requerido para obtener resultados.

Parámetros del modelo de validación cruzada

Para este caso específico se utilizó un **K=5** y se repitió el procedimiento 5 veces para cada algoritmo utilizado. Además se decidió generar una submuestra (por eficiencia computacional) de 4000 noticias seleccionadas aleatoriamente del *Corpus* de los cuales 1991 fueron catalogados previamente en la categoría "LAFT" y las restantes 2009 noticias fueron catalogadas en la categoría "Otros temas". Adicionalmente otros 1199 documentos totalmente externos al procedimiento de validación cruzada fueron separados para una validación totalmente externa al proceso de modelado.

3.3-Depuración de las variables utilizadas para los modelos

Una etapa importante en la construcción de modelos predictivos es la llamada "selección de características", en la cual se analizan las variables predictoras para medir su grado de importancia con respecto a la variable dependiente y decidir si se deben eliminar o mejorar algunas variables. En este caso, se utilizó el *Odds ratio* de pertenecer a la clase "LAFT" sobre la clase "Otros", de tal manera que

todos los *tokens* que se encontraran con valores de *Odds Ratio* cercanos a un(los valores escogidos fueron entre 0.5 y 2) fueron eliminados para lograr un modelo parsimonioso pero sin sacrificar la precisión.

3.4-Métricas utilizadas para medir el error

En el ámbito del Procesamiento del Lenguaje Natural las métricas de precisión de los modelos tienen ciertas particularidades de interpretación con respecto a otras aplicaciones más tradicionales, aunque los cálculos sean similares. A continuación se detallan las métricas utilizadas (Powers, 2007):

Recall

Esta métrica responde a la pregunta ¿Cuántos documentos relevantes (clase LAFT) son seleccionados? Es decir, de todos los documentos LAFT verdaderos, que proporción fue clasificada como LAFT por el modelo. Este indicador mide que tan efectivo es el modelo para captar todos los casos de noticias que realmente son LAFT, el remanente serán casos nunca analizados por haber sido considerados como falsos negativos. También es conocida en aplicaciones más generales como "sensibilidad". Se calcula de la siguiente forma:

$$\textbf{Recall} = \frac{\mathrm{VP}}{\mathrm{VP} + \mathrm{FN}}$$

Donde **VP=** Verdaderos positivos y **FN=** Falsos negativos.

Precisión

Esta métrica responde a la pregunta ¿De los documentos clasificados como clase LAFT, cuántos eran realmente clase LAFT? Este indicador mide la proporción de documentos que serán realmente clase LAFT en una futura implementación y el remanente serán falsos positivos que deberán obviar los analistas. También es conocida como "Valor de Predicción Positivo". La fórmula de cálculo es la siguiente:

$$\text{Precision} = \frac{\text{VP}}{\text{VP} + \text{FP}}$$

Donde FP= Falsos Positivos.

F1

Es el resultado de la media armónica entre los dos indicadores anteriores y que brinda un punto intermedio en la interpretación y un parámetro más general para la selección de modelos:

$$\text{F1} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Curva ROC

La curva ROC es un enfoque mucho más general, representa gráficamente la sensibilidad o Recall frente a 1- Especifidad (proporción de negativos que son correctamente clasificados). Los modelos que tengan una mayor área bajo la curva ROC, son considerados los mejores, ya que para el juego de modelos que

utilizados para comparar, son los que poseen mayores valores de sensibilidad y especifidad.

3.5- Datos de validación externa

Además de la etapa de pruebas para medir el error de los modelos escogidos, se realizó una prueba de concepto para verificar que con datos totalmente ajenos al proceso de modelado, el resultado final sea coherente y funcione en posibles aplicaciones reales. Para esto se seleccionó una lista de individuos conocidos por su involucramiento en temas de LAFT así como individuos reconocidos por la opinión pública, pero que al momento de la realización de este trabajo no tenían relación con temas LAFT. Para estos individuos se extrajo una muestra de las más recientes noticias y cada una de estas fue calificada por el modelo.

Resultados

La sección de resultados se divide en el análisis exploratorio que se realizó antes de iniciar con los modelos para poder conocer mejor los datos y ajustar detalles de ruido en estos para el modelado (que ya fueron explicados en el apartado metodológico en la sección de *Text Analytics*).Este análisis exploratorio se realizó con los mismos datos extraídos para el modelado, pero acotándolos en algunos casos a Latinoamérica o Centroamérica. Posteriormente se muestra el análisis de selección de características para reducir y mejorar las variables que entrarían a los modelos, luego el entrenamiento de los modelos y la prueba de los mismos en el

set de datos de validación. Finalmente se muestran los resultados de la prueba de concepto.

Análisis Exploratorio

Una primera visualización clásica en la minería de texto, es el *wordcloud* en el cual se muestran los términos más relevantes en la muestra obtenida de la categoría LAFT, según el **Tf-Idf:**

Gráfico 3: Word Cloud de Tokens más importantes

De forma clara se muestran las palabras más importantes según el tamaño que ocupan en el gráfico, que incluye *tokens* de una o dos palabras, como por ejemplo *drug, mariguana, pólice, bitcoin o cocaine*. Este análisis sirvió para encontrar *stop Words* adicionales y eliminarlas de los modelos.

Otros análisis complementarios realizados para explorar objetivos exploratorios del estudio fue el seguimiento de las noticias por áreas geográficas y a través del tiempo, para medir tendencias e impacto del tema LAFT en los países donde puede tener presencia un conglomerado bancario latinoamericano.

Con este tipo de análisis se puede llevar el pulso de la frecuencia de estas noticias por países y determinar donde se están concentrando, tanto en términos absolutos como relativos para un momento específico. El gráfico 4 ejemplifica este resultado.

Gráfico 4: Latinoamérica, Frecuencia de Noticias por país (Mayo-Junio 2015)

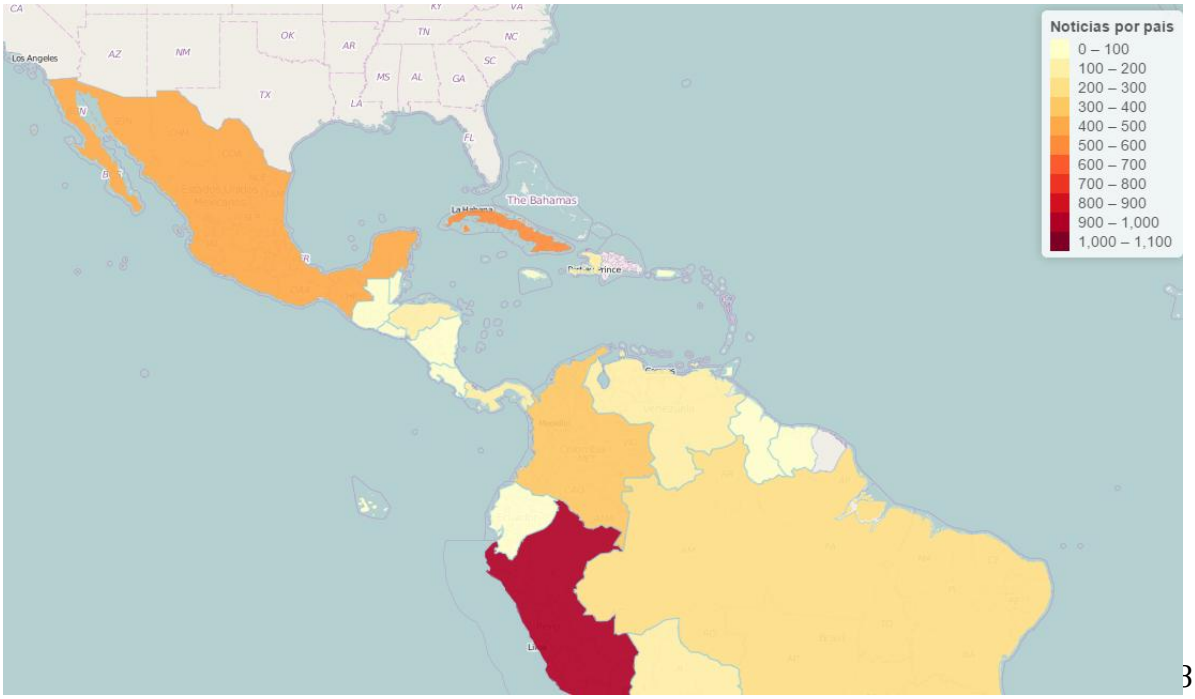
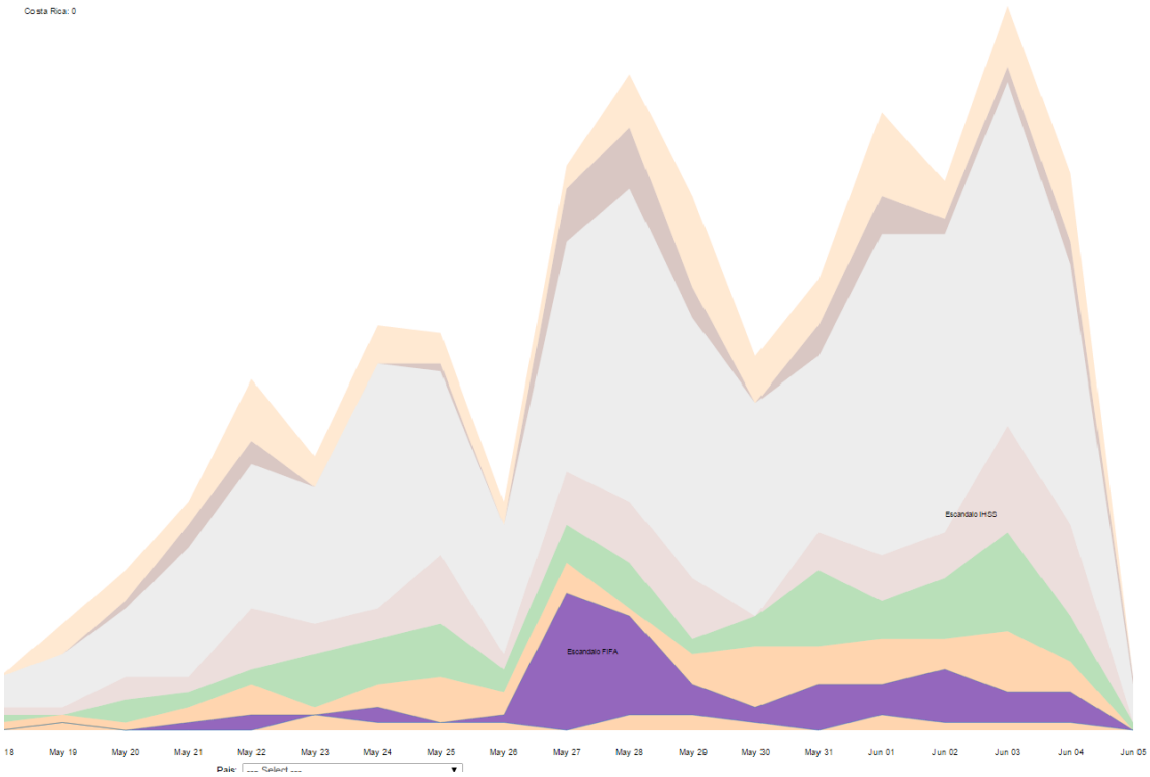


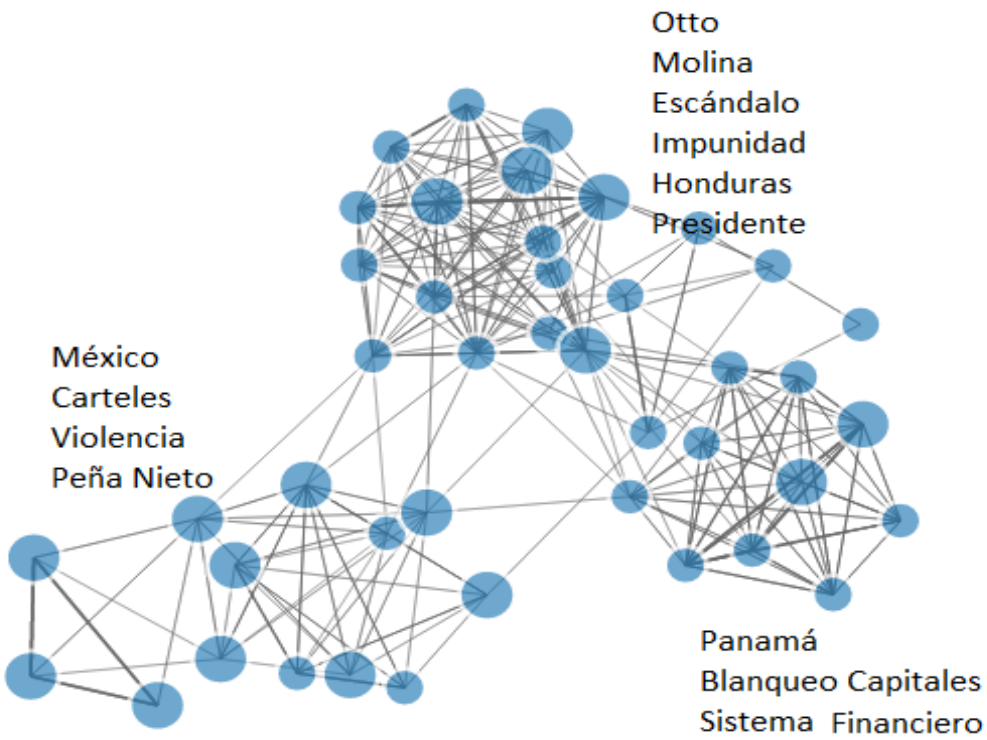
Gráfico 5: Centroamérica y México, cantidad de noticias por país (Mayo-Junio 2015)



Además, como muestra el gráfico 5, se puede medir la tendencia de la frecuencia de las noticias por país a través del tiempo, para identificar ciertos eventos ocurridos asociados al tema. Por ejemplo, para la muestra recogida, se puede observar el impacto que tuvo el escándalo de FIFA en Costa Rica y los escándalos de corrupción de gobernantes en Honduras.

Por último, se pueden asociar temas específicos con los países, para conocer cuál es el tipo de problema asociado a lavado de activos que caracteriza a cada uno de los países. El gráfico 6 muestra un ejemplo.

Gráfico 6: Relaciones entre *tokens* con países

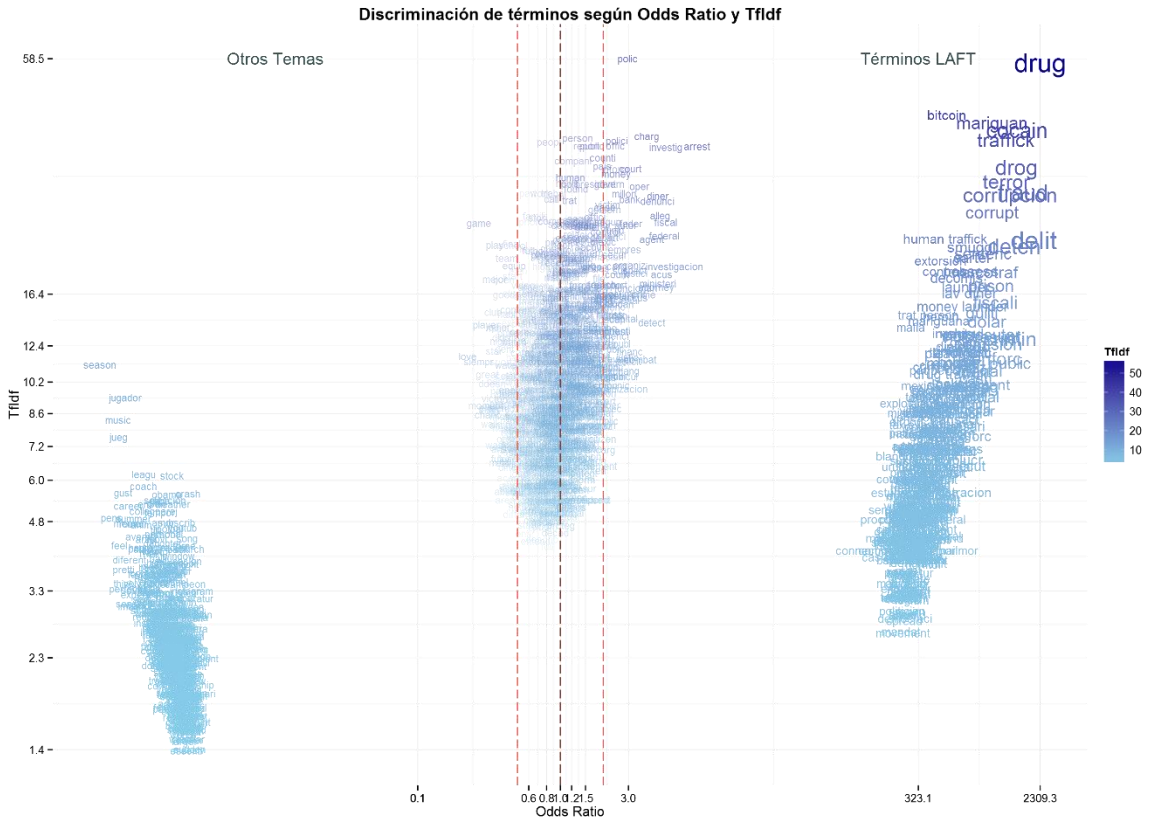


Selección de características

Debido a la gran cantidad de *tokens* que inicialmente ingresarían a los modelos como variables predictoras, se realizó un procedimiento para eliminar aquellas términos que no discriminen entre los textos de clase LAFT y el resto de noticias y dejar solo aquellos con mayor poder predictivo. Para esto se obtuvo el *Odds ratio* de que dado un *token*, este pertenezca a un texto de clase LAFT. El gráfico 7 muestra la relación entre el *Odds ratio y el Idf-Tf* por cada uno de los *tokens*, en donde se aprecia según el tamaño las variables de mayor poder discriminatorio.

Los términos con valores cercanos a uno en su *Odds ratio* (nube de puntos central), son los candidatos a eliminar del modelado.

Gráfico 7: Discriminación de términos según *Odds Ratio* y Tf-Idf



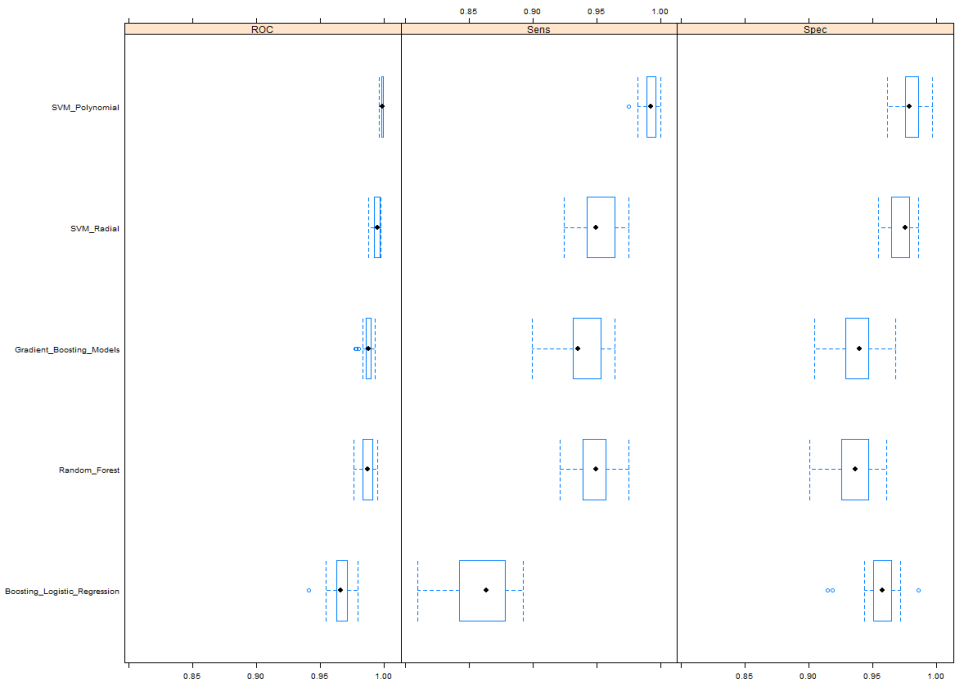
Se decidió eliminar todos los términos con un *Odds ratio* entre 0.5 y 2.

Modelado

Comparativa de modelos

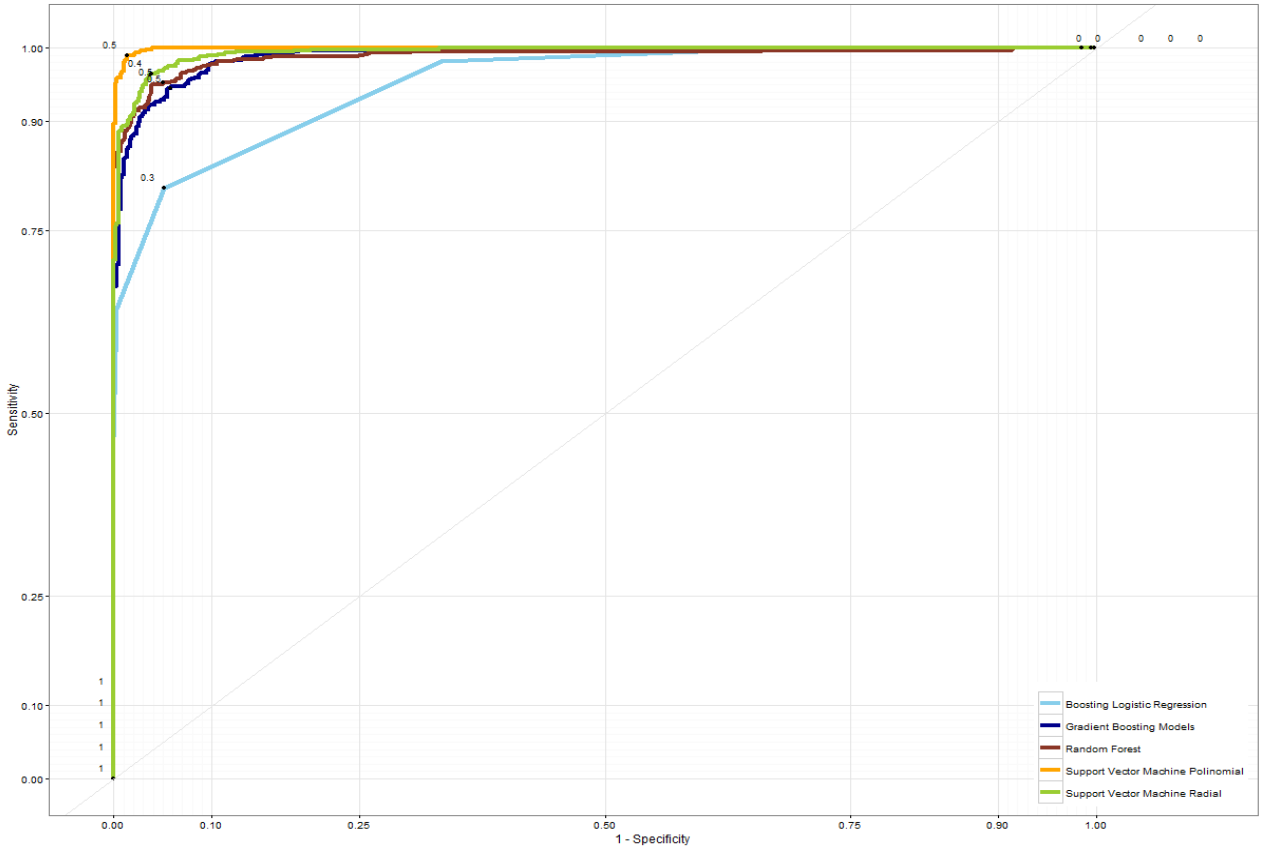
Cada uno de los algoritmos de modelado estadístico fueron ejecutados con la configuración de validación cruzada reportada en la sección metodológica para estimar cuál combinación de valores en cada algoritmo generaba el menor error de clasificación en la etapa de entrenamiento. El modelo que en cada algoritmo arrojara mejores resultados se probó en la tabla de prueba, obteniendo los siguientes resultados.

Gráfico 8: Resultados de validación cruzada de modelos según métricas de efectividad



En este gráfico se comparan tres métricas básicas (ROC, Sensitividad y Especifidad) para la evaluación de la efectividad de los cinco modelos tomados en cuenta. Para la curva ROC, todos los modelos tienen valores mayores a 95%, pero los modelos SVM tanto en su versión de kernel polinomial como radial cuentan con valores de 99%. En las otras dos métricas todos los modelos tienen rendimiento superior al 85%, pero de igual forma los SVM siguen con el menor error.

Gráfico 9: Comparación de Curvas ROC por modelo



Realizando un *zoom* a la curva ROC, se puede observar como cuatro modelos tiene rendimientos muy similares, siendo levemente superior el Support Vector Machine con kernel polinomial. El *Random Forest*, Support Vector Machine radial y el *Gradient Boosting* tienen una curva muy similar. El modelo *Boosted Logistic Regression* se separa de los otros cuatro modelos al tener un rendimiento inferior.

Para resumir los resultados de las métricas más importantes en el contexto del Procesamiento de Lenguaje Natural (Recall, Precision y F1) y ordenar los modelos de acuerdo a su rendimiento en estas, se detalla el resultado final en la siguiente tabla:

Cuadro 4: Métricas de rendimiento de modelos en tabla de testing

Modelo	Recall	Precisión	F1
Random Forest	95.8	93.8	94.8
Boosted Logistic Regression	80.9	94.0	87.0
Support Vector Machine(Radial)	95.8	96.3	96.0
Support Vector Machine(Polinomial)	99.0	98.3	98.6
Gradient Boosting Models	94.0	94.4	94.2

Consenso de modelos

Debido a que todos los modelos se desempeñan de manera adecuada y diferenciada según la naturaleza del texto, se decidió realizar un consenso de estos modelos por medio de una regresión logística, para obtener los coeficientes

de cada uno y darles una ponderación. Este modelo de consenso mejoró aún más las métricas de rendimiento en la tabla de prueba.

Cuadro 5: Modelo de consenso

Modelo	**Coeficiente**
Random Forest	10.71
Support Vector Machines Polinomial	9.55
Support Vector Machines Radial	0.42
Boosted Logistic Regression	-1.2
Gradient Boosting Models	-2.19
Intercepto	-9.1419
Grados de libertad: 1198 / Residual Deviance: 69.76 / AIC: 81.7	

Para el modelo de consenso se obtiene una leve mejora en los resultados:

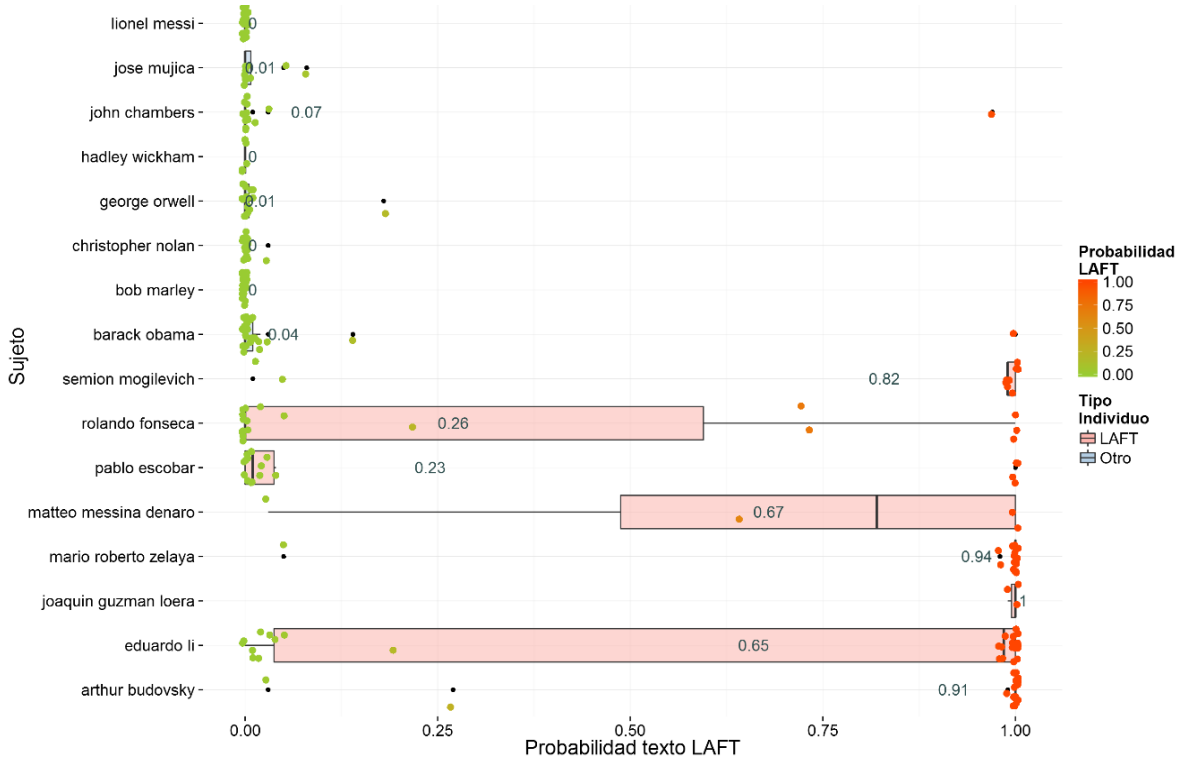
1. Tasa de acierto de 98.83 %.
2. **Precisión** (Documentos clasificados LAFT que son verdaderamente LAFT) de 98.83 %.
3. ***Recall*** (De todos los documentos LAFT, cuántos son clasificados LAFT) de 98.83 %.

   Lo anterior indica que si esta aplicación se ejecuta realmente, se esperaría un 1,17% de falsos positivos y detectaría el 98,8% de todos las noticias LAFT bajo un *query* especificado.

Validación externa

Para validar uno de los objetivos centrales de la aplicación, que es evaluar la probabilidad que tienen las noticias de la web de un individuo o entidad en específico a pertenecer a temas LAFT, se escogieron ciertos individuos que no destacan en temas relacionados con LAFT pero sí con otros temas (deportes, ciencia, política, farándula) e individuos que ya son reconocidos por su relación con el lavado de activos o la financiación del terrorismo, con el fin de verificar la efectividad de la aplicación.

Gráfico 10: Distribución de la probabilidad de textos LAFT por cada sujeto



Este gráfico muestra la distribución de cada individuo consultado, en función de la probabilidad estimada por el modelo de consenso de cada noticia. A grandes rasgos se observa cómo los individuos de clase "Otro", tienen distribuciones con valores muy bajos en comparación con los individuos de clase "LAFT", aunque existen ciertos puntos (noticias) con valores altos para individuos que no están relacionados al tema LAFT. Por otro lado cabe destacar los valores del individuo Rolando Fonseca, exjugador de fútbol costarricense que luego de su carrera deportiva fue asociado a ciertos negocios de lavado de activos y de ahí que un grupo de noticias haya sido calificado con valores bajos y otro con valores altos. Esta misma característica la tiene Eduardo Li, ex directivo de fútbol costarricense que luego fue acusado por actos de corrupción dentro de la FIFA. El caso de Pablo Escobar, en el que el score o probabilidad de la mayoría de las noticias es bajo, se constató debido a que se trata de un personaje que hoy día trasciende el área de lavado de activos y se asocia a muchas noticias de la farándula actualmente, las cuáles son clasificadas correctamente por la aplicación.

Discusión

En este documento se describe cómo se logra diseñar exitosamente un prototipo para extraer, estructurar, analizar y sacarle provecho a la información pública que se encuentra en la web. Este tipo de información y modelado es aún poco utilizado por las empresas del sector financiero pero una vez que se detalla una metodología para su explotación se puede sacar valor para potenciar el negocio y

generar conocimiento que de otra forma pasaría desapercibida o tomaría muchos recursos descubrirla.

En este caso, la metodología tiene muchas usos en los departamentos de cumplimiento de organizaciones de intermediación financiera, ya que a partir de esta se pueden desarrollar aplicaciones que complementen y potencien las herramientas de monitoreo disponibles.

El principal uso que el autor encuentra en este prototipo, es el desarrollo de una aplicación para que todos los clientes o gestores relacionados a una organización financiera sean calificados por los algoritmos para medir de que en caso de que tengan noticias en la web, estas estén asociadas a temas LAFT y generar alertas para los clientes que tengan scores elevados, en un proceso de *batch* automático (corrida masiva) o de consulta por parte de los analistas mediante interfaz web gráfica, similar a los sistemas que chequean a los individuos para cruzar con listas de vigilancia. No obstante, los usos no se limitan al ejemplo anterior, ya que toda la información capturada puede ayudar a llevar un monitoreo más integral de la realidad y tendencias de lavado de activos y financiación de terrorismo en un área geográfica determinada, en un espacio de tiempo dado.

La aplicación debe ser mejorada por medio de la retroalimentación de los analistas, ya que como cualquier aplicativo de clasificación automático se encuentra susceptible de mejora continua, el modelado debe ser ajustado para considerar estos *inputs*. A su vez, los modelos deben ser entrenados periódicamente con datos nuevos, para capturar las nuevas tendencias en lavado

de activos y financiación del terrorismo y así no sesgarse por temas que estén desactualizados. Se debe tomar en cuenta que esta aplicación fue generada con los datos extraídos entre abril y junio del año 2015, por lo que los modelos podrían estar influenciados por los eventos sucedidos en ese periodo. Para evitar una influencia muy alta de eventos aislados en un momento del tiempo dado, se deberían entrenar los modelos con un periodo de tiempo mayor como mínimo un año de historia.

Los procesos de cumplimiento para tratar escándalos como los sucedidos en el año 2016 como *Panamá Papers* (que no fueron evaluados en los datos de esta aplicación), podrían verse altamente beneficiados por esta aplicación, en el entendido de que debe encontrarse debidamente automatizada.

Referencias

Annau, M. (10 de Mayo de 2015). Short Introduction to tm.plugin.webmining.

Atre, S., & Blumberg, R. (2003). The Problem with Unstructured Data. *dmreview*, 42-46.

Breiman, L. (1996). Bagging predictors. *Machine Learning*.

Buchta, C., Feinerer, I., Geiger, W., Hornik, K., Mair, P., & Rauch, J. (2013). The textcat Package for n-Gram Based Text Categorization in R. *Journal of Statistical Software*.

Buhlmann, P., & Dettling, M. (2002). Boosting for tumor classification with gene expression data. *Bioinformatics*, 1061-1069.

Feinerer, I., Hornik, K., & Meyer, D. (07 de Julio de 2008). Text Mining Infrastructure in R. *Journal of Statistical Software*.

Friedman, J. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*.

Friedman, J., Hastie, T., & Tibshirani, R. (2009). *Elements of Statistical Learning.*

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. Stanford.

Lang, D. T. (4 de Agosto de 2004). Word Stemming in R.

McAfee, A., & Brynjolfsson, E. (2012). Big Data: The Management Revolution. *Harvard Bussiness Review*, 60-70.

Powers, D. (2007). *Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation.*

Ramos, J. (2003). Using TF-IDF to Determine Word Relevance in Document Queries.