Análisis de contenido político en medios de comunicación en México mediante técnicas de NLP

Alfredo Lozano, Augusto Sagaón

Instituto Tecnológico Autónomo de México

15 de diciembre de 2017

Resumen

El propósito de este artículo es realizar un análisis sobre el contenido de medios de comunicación en México aplicando distintas técnicas de aprendizaje de máquina que nos permitan dar una buena aproximación al problema de detectar el sesgo en ciertos medios. En la primera parte explicaremos la forma en que abordaremos el problema, introduciremos a la base de datos y explicaremos el pipeline de pre-procesamiento que tuvimos que seguir mediante técnicas de procesamiento de lenguaje natural, después haremos un primer ejercicio de clustering no supervisado cuyo resultado es un punto determinante en el proyecto y a partir de ahí elaboraremos sobre la construcción e interpretación de un modelo supervisado de clasificación sobre la inclinación política de las fuentes.

Índice

1. Motivación

Es un problema muy conocido en las ciencias sociales, que muchas de las investigaciones sobre los medios centran su atención en detectar el sesgo de las noticias sobre temas particulares como política, la migración, las guerras o racismo. En estas investigaciones los artículos de noticias son principalmente seleccionados y analizados manualmente usando un proceso de codificación o mediante marcos teóricos como el análisis del discurso y el análisis de contenido.
Es común el pensar que, respecto a política, hay medios de comunicación que se inclinan mas por ideologías políticas definidas que denominamos de "derecha" mientras que otros medios se les identifica con ideologías de "izquierda". Vamos a enfocar nuestro interés en hacer un aproximamiento a esta creencia usando distintas técnicas de aprendizaje de máquina para analizar encabezados de medios de comunicación escrita con presencia en internet[1].
Por lo tanto, desde un enfoque computacional, revisaremos algunos de los métodos para el procesamiento de lenguaje natural [6] con la ventaja de poder procesar una gran cantidad de datos para detectar estos patrones de sesgo.
De aquí en adelante nos vamos a concentrar en el nivel semántico del lenguaje, esto es, nos interesan la presencia de ciertas palabras y ciertos grupos de palabras en los documentos (o encabezados) en un sentido frecuentista e ignoramos las relaciones gramaticales entre las mismas, tomamos este supuesto en base a que el lenguaje en cada encabezado debe ser concreto y directo para atraer la atención del lector, entonces creemos que la semántica juega un papel más importante, aunque un análisis gramático sería una buena forma de complementar este análisis.

1.1. Descripción de la base de datos

Para comenzar, generamos una base de datos de los encabezados de las noticias presentadas por los medios de cobertura nacional rankeados en los sitios visitados de México según el sitio web Alexa en 2015.

Una observación de nuestros datos consta de: fecha, encabezado, fuente (medio que publicó la noticia), una bandera booleana que representa si el encabezado está completo y el resultado del pre-procesamiento del encabezado original.

	fecha	encabezado	fuente	link	incompleta	enc_prep
13	2017-10-18	renuncia mariana moguel al pri-cdmx	eluniversal	http://www.eluniversal.com.mx/metropoli/cdmx/r...	False	renunciar mariano moguel pricdmx
14	2017-10-17	definirá pri método de selección de candidatos...	eluniversal	http://www.eluniversal.com.mx/nacion/politica/...	False	definir pri método selección candidato viernes
15	2017-10-15	redestape de presidenciables del pri	eluniversal	http://www.eluniversal.com.mx/columna/periodis...	False	redestape presidenciable pri
16	2017-10-19	¿quién es el tapado del pri? las señales en lo...	eluniversal	http://www.eluniversal.com.mx/columna/carlos-i...	False	¿quién tapar pri señal pino
17	2017-10-17	el pri tiene su propia cultura para definir ca...	eluniversal	http://www.eluniversal.com.mx/nacion/politica/...	False	pri propio cultura definir candidatos epn

Figura 1: Base de datos

Para recolectar las noticias que usamos para este reporte, creamos un script en Python que se ejecuta automáticamente todos los días a las 10:30 a.m. desde el día 14 de Octubre del 2017 y juntamos encabezados hasta el día 12 de Diciembre del 2017. En este periodo de tiempo, se juntaron 19405 noticias con las características especificadas, para las palabras clave y las fuentes (por separado) siguientes:

[1] **Disclaimer:** No es de nuestro interés manifestar nuestra propia inclinación política, sino que creemos que investigación (imparcial) en este campo es necesaria para complementar el contenido periodístico.

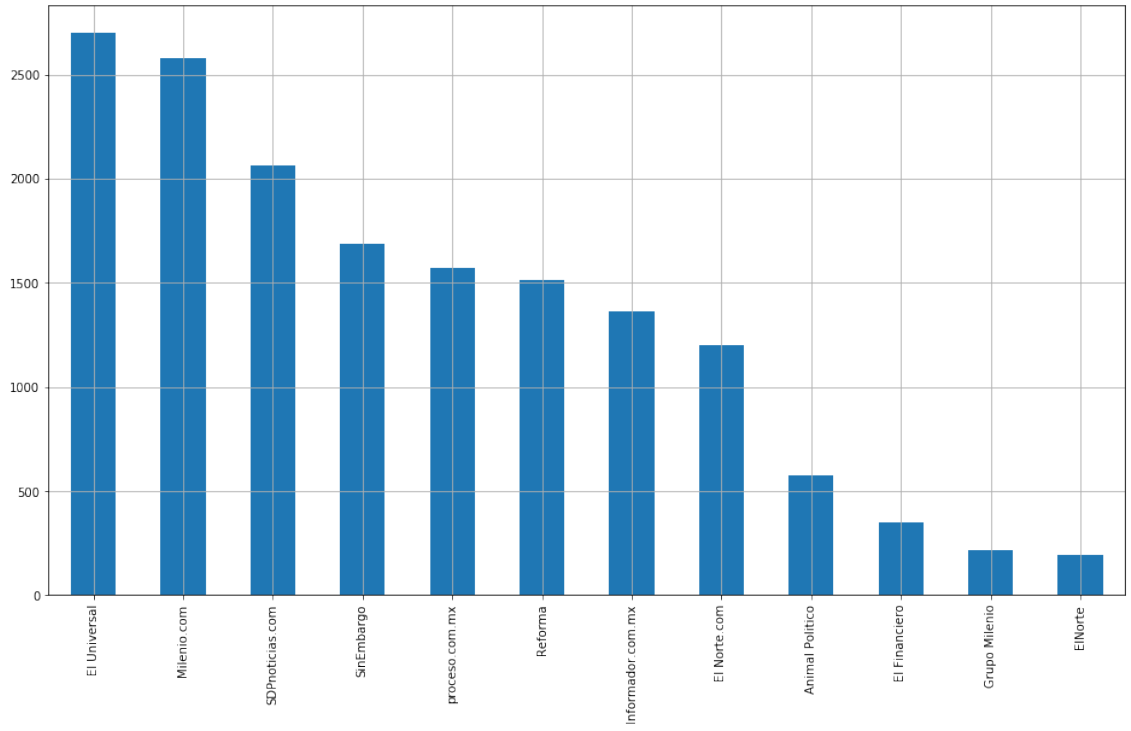

Figura 2: Histograma: fuente de los encabezados

Palabras clave:

- PRI
- PAN
- PRD
- MORENA
- MARGARITA
- ZAVALA
- AMLO
- ANDRES MANUEL
- LOPEZ OBRADOR
- OSORIO CHONG
- MARICHUY
- JAIME RODRIGUEZ
- RODRIGUEZ CALDERON
- EL BRONCO
- MEADE
- NARRO
- ERUVIEL
- ANAYA
- EPN
- ENRIQUE PENA NIETO
- MIGUEL ANGEL MANCERA
- MANCERA

Medios :

- www.eluniversal.com.mx
- www.proceso.com.mx
- www.excelsior.com.mx
- www.informador.com.mx
- www.reforma.com
- www.animalpolitico.com
- www.sinembargo.mx
- www.milenio.com
- www.elnorte.com
- www.sdpnoticias.com

2. Pre-procesamiento de la base

Llevamos a cabo un pipeline de pre-procesamiento de los datos en el que primero pasamos todo el texto a minúsculas, removimos caracteres especiales y puntuación, después mapeamos términos que nos interesan a un diccionario establecido (un ejemplo: todas las ocurrencias de “lopez obrador“

y “andres manuel“ las mapeamos a “amlo“), removimos las *stop words* (que son palabras específicas del lenguaje como artículos y preposiciones), por último definimos un *stemming* que mapea palabras como “corrido“ y “corriendo“ a una raíz “correr“ en base a una lista de Lematización definida en [1].
Como resultado un encabezado como “ el pri y el frente adelantan la guerra electoral“ se traduce en: “pri frente adelantar guerra electoral“.

Sin embargo, esto no es suficiente, la forma en la que vamos a introducir los datos al clasificador es mediante la representación $tf - idf$ (term frequency - inverse document frequency) de nuestra matriz de datos que representa la frecuencia relativa de cada palabra en un documento y en el corpus entero, para esto necesitamos tokenizar los encabezados (verlos como la lista de las palabras que les conforman) y después calculamos la matriz.

2.1. $tf - idf$

Tomemos como corpus al conjunto de todos los términos (o palabras) en los documentos (o encabezados) que estamos analizando y a cada encabezado como un documento independiente, la matriz de tf-idf es una técnica de procesamiento de lenguaje natural que hace uso de la frecuencia relativa de cada término con respecto de cada documento y con respecto del corpus.

Para cada término en un documento calculamos su tfidf como sigue:

$$tf\_idf_t = tf \cdot log_{10}\frac{N}{df}$$

Donde:

- tf: Frecuencia del término en el corpus
- N: Número de documentos en el conjunto de datos
- df: Número de documentos que contienen el término

Primero vamos a tokenizar los encabezados procesados, para esto generamos una lista con las palabras de cada documento de la siguiente manera:

```
enc_prep:  edomex integrar consejo electoral municipal  distritales
tokens: ['edomex', 'integrar', 'consejo', 'electoral', 'municipal', 'distritales']

enc_prep: venezuela   pri
tokens: ['venezuela', 'pri']

enc_prep: eruviel sacará  flote  pri  cdmx gutiérrez   torrar
tokens: ['eruviel', 'sacará', 'flote', 'pri', 'cdmx', 'gutiérrez', 'torrar']

enc_prep:  pri expulsar  alcalde  macuspana  denuncia  corrupción
tokens: ['pri', 'expulsar', 'alcalde', 'macuspana', 'denuncia', 'corrupción']

enc_prep: medio mexicano  adicto  dinero público  poder ayudar
tokens: ['medio', 'mexicano', 'adicto', 'dinero', 'público', 'poder', 'ayudar']
```

Figura 3: Tokenización de los encabezados

Luego calculamos frecuencia de cada término:

```
fuente : proceso
top 10 keywords: [('“', 134), ('epn', 115), ('amlo', 113), ('pri', 99), ('zavala', 81), ('meade', 73), ('pan', 73),
('mil', 61), ('pedir', 57), ('frente', 55)]
---
fuente : eluniversal
top 10 keywords: [('amlo', 219), ('pri', 187), ('meade', 173), ('pan', 151), ('anaya', 140), ('epn', 139), ('morena',
125), ('prd', 125), ('frente', 119), ('zavala', 94)]
---
fuente : sdpnoticias
top 10 keywords: [('amlo', 234), ('meade', 203), ('pri', 174), ('epn', 131), ('frente', 108), ('anaya', 107), ('zaval
a', 100), ('morena', 94), ('candidato', 94), ('pan', 90)]
---
fuente : sinembargo
top 10 keywords: [('pri', 125), ('“', 123), ('amlo', 106), ('epn', 95), ('decir', 92), ('méxico', 89), ('meade', 68),
('pan', 61), ('dar', 58), ('frente', 55)]
---
fuente : animalpolitico
top 10 keywords: [('pri', 66), ('decir', 65), ('méxico', 59), ('gobernar', 56), ('cdmx', 47), ('epn', 46), ('mdp', 4
5), ('sismo', 43), ('año', 42), ('ir', 41)]
---
```

Figura 4: Frecuencia de términos

Por último calculamos los valores $tf - idf$ de cada término:

	tfidf
amlo	3.615314
pri	3.672131
meade	3.864969
epn	3.887611
pan	4.125655
frente	4.235600
méxico	4.288317
morena	4.296901
anaya	4.299778
zavala	4.305559

Figura 5: Términos con tfidf más bajo

	tfidf
ayer hoy	7.751208
razón	7.751208
audiencia	7.751208
muerto	7.751208
aurelio	7.751208
difundir	7.751208
operación	7.751208
aventajar	7.751208
crítica	7.751208

Figura 6: Términos con tfidf más alto

Es interesante ver que en el sentido $tfidf$ palabras con un valor bajo, son las palabras que ocurren con mayor frecuencia en el corpus y, en cambio, le damos mayor peso para la generación de contexto a palabras con una menor frecuencia.

2.2. Latent Semantic Analysis

Para reducir el trabajo computacional de entrenar nuestros algoritmos queremos reducir la dimensionalidad y además mantener representada la mayor variabilidad posible de los datos, por lo que aplicamos *Latent Semantic Analysis* que es una transformación similar a *Principal Components Analysis* que se utiliza para reducir la dimensión de la base de datos en términos de la descomposición en valores singulares (SVD). Empleamos esta técnica porque tiene ventajas con respecto de PCA cuando la matriz de datos es rala.

3. Visualización dos-dimensional

Ahora, algo que nos interesa en nuestro estudio es poder generar una visualización que nos permita interpretar nuestros algoritmos, para esto utilizaremos el *Stochastic Neighbor Embedding*.

3.1. Stochastic Neighbor Embedding

El SNE es un algoritmo que mapea objetos de alta dimesionalidad en objetos de poca dimensionalidad, preservando la estructura de parentesco de una vecindad, sin importar el intercambio que resulta de una clasificación incorrecta alrededor de objetos lejanos. El SNE no tiene nada que ver con las medidas de similitud pero está basado en los conceptos de entropía y divergencia probabilística.

La idea principal es centrar una distribución normal para cada valor de entrada (a lo largo del espacio de alta dimensionalidad) a fin de usar su densidad para definir una distribución de probabilidad de todos los vecinos. El objetivo es aproximar esta distribución de probabilidad tanto como sea posible replicando la estructura de parentesco en un espacio de baja dimensionalidad.

4. Clustering

Decidimos empezar la labor analítica haciendo clustering con *KMeans* para detectar algunos patrones en los datos y como herramienta que aumente el poder predictivo.

Entrenamos el algoritmo con $k \in \{3, 10, 20\}$ donde k es el número de clusters a realizar mediante distancia euclidiana, escogimos estos valores para tener una idea del nivel de homogeneidad que podemos obtener a partir del enfoque $tf - idf$.

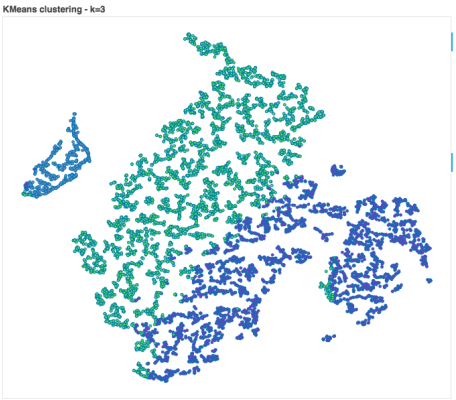

Figura 7: $k = 3$

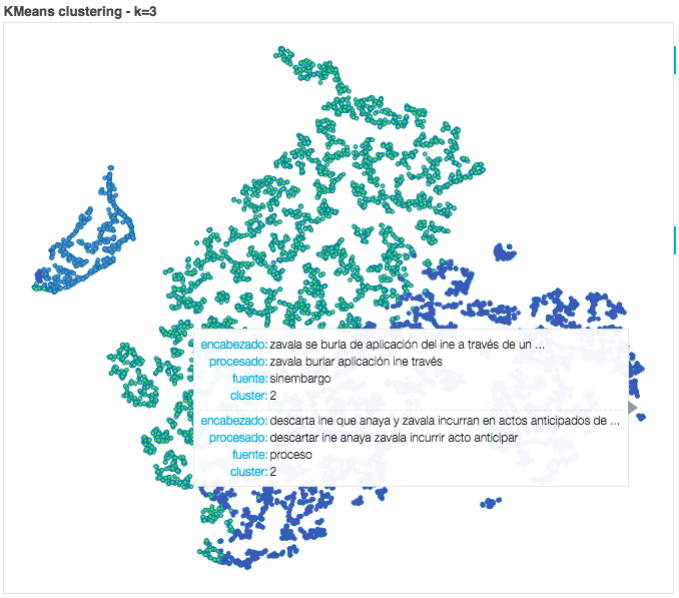

Figura 8: $k = 3$ ejemplo de miembros

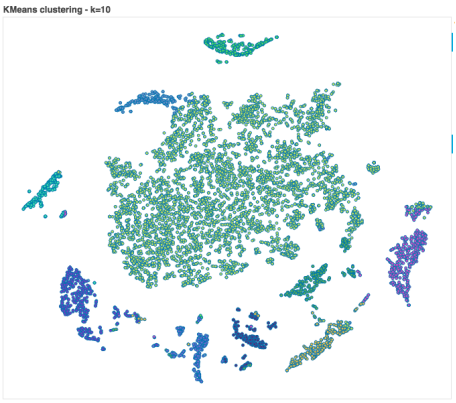

Figura 9: $k = 10$

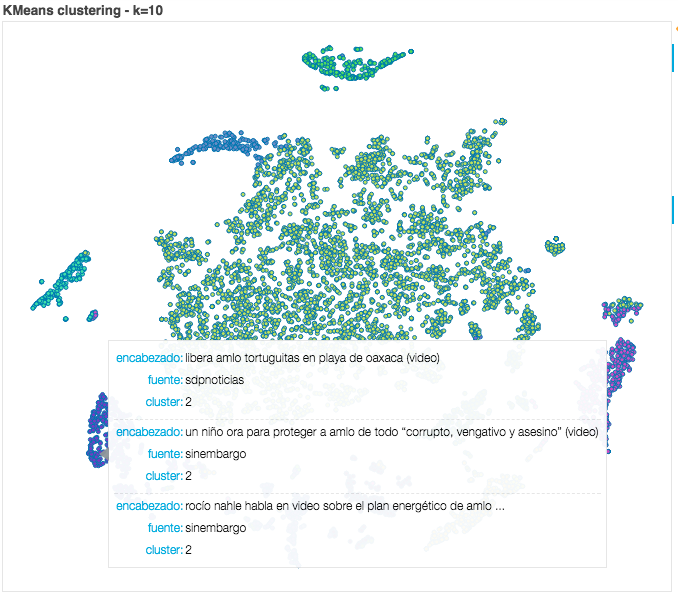

Figura 10: $k = 10$ ejemplo de miembros

Figura 11: $k = 20$

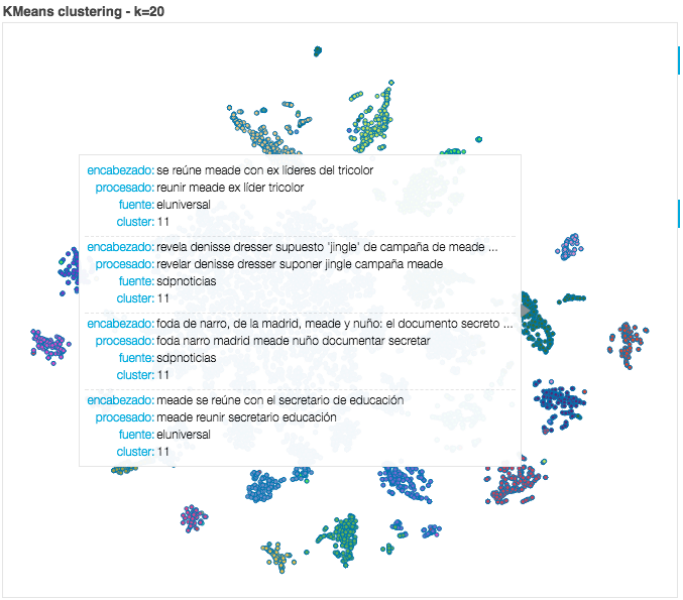

Figura 12: $k = 20$ ejemplo de miembros

Era de esperarse que a manera que aumenta k los clusters se vuelven más homogéneos, en general notamos que los clusters generados si mantienen un muy buen nivel de segmentación, incluso para una base de datos que si bien tiene encabezados restringidos a partidos y actores políticos también tiene encabezados generales de cada uno de los medios que nos interesa y esto es muy evidente en los clusters generados, vemos que en los tres casos hay un cluster muy grande en el centro que representa las noticias más generales y pequeños clusters que le rodean en los que están agrupadas muchas de las noticias restringidas a estos keywords que nos interesan.

5. Clasificación

Ahora bien, si lo que nos interesa es detectar de qué manera los encabezados de las noticias que publica un medio nos indican si este medio tiene alguna inclinación política, queremos entrenar un algoritmo que tenga como salida la inclinación política de un encabezado.
Siguiendo con esta idea vamos a proceder por entrenar un modelo de *Random Forest* usando *Grid Search* para seleccionar los hiperparámetros.

Dado que queremos usar un método de aprendizaje supervisado, necesitamos etiquetar las noticias, por lo que partimos del siguiente supuesto, basado en opinión pública: los medios "el Universal" y "SDP noticias" tienden a favorecer a las acciones de partidos de "derecha" y los medios "Proceso", "Animal Político" y "Sinembargo" son mayormente asociados con partidos de "izquierda". Basados en este supuesto, ya tenemos una variable target que predecir, pues convertimos la fuente de la noticia en binaria, si el partido se asocia comunmente con partidos de "derecha" le asignamos un 1, y un 0 en otro caso. Para este ejercicio solo nos concentraremos en las fuentes mencionadas.

Una vez construida la matriz de información *tf-idf*, le aplicamos el método de *SVD* para reducir la dimensionalidad y le anexamos la información del número de cluster al que fue asignado cada encabezado y esta fue nuestra matriz X de features, mientras que el vector y de target fue la columna que indicaba si el medio era asociado con la "derecha" o "izquierda".

5.1. Random Forest

Posteriormente, separamos nuestros datos en subconjuntos de *train* y *test*, usando un a proporción de 3 a 1. Una vez hecha esta separación, procedimos a hacer una busqueda de hiperparámetros usando la función de sklearn *GridSearchCV* que prueba las combinaciónes de hiperparámetros

que definimos y hace $Cross - Validation$ en cada prueba. Los mejores parámetros encontrados para este problema fueron, de los que modificamos, $max\_features = log2, n\_estimators = 100, max\_leaf\_nodes = None$. A continuación presentamos algunas métricas del modelo, usando la base de datos completa

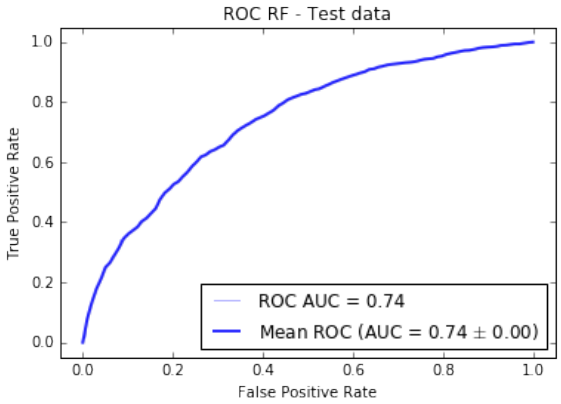

Figura 13: Curva ROC

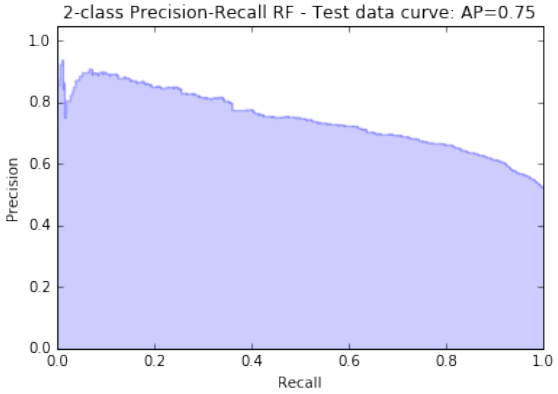

Figura 14: Precision-Recall

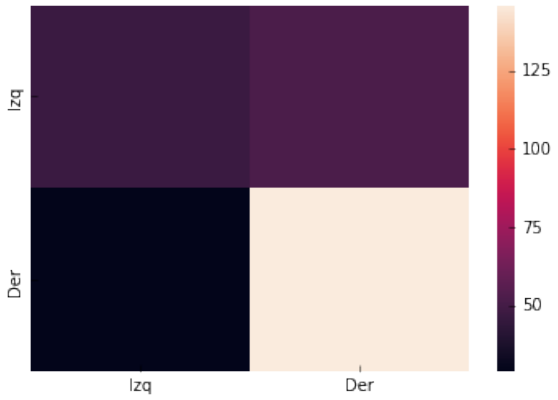

Figura 15: Matriz de confusión

Podemos notar que el modelo parece si estar aprendiendo algo sobre las variables que proporcionamos, pero no es un desempeño que consideraremos suficiente para poder hacer alguna clasificación precisa. Esto era de esperarse pues dentro de esta base de datos debe haber muchos encabezados que sean imparciales, por lo que a continuación exploraremos el reducir el número de temas y observar como cambia el desempeño del modelo.

5.2. Reduciendo temas

Hicimos el mismo procesamiento para un subconjunto de nuestra base de datos, restringiendo solo a noticias con las palabras “amlo“ y “meade“, bajo el supuesto de que son temas polarizantes entre medios de inclinaciones contrarias.
A continuacion mostramos los resultados de la visualización de los clusters de esta base, probando con 3, 10 y 20 clusters.

Figura 16: $k = 3$

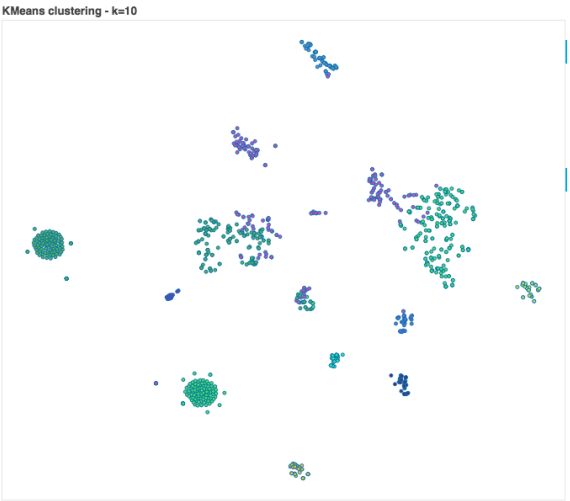

Figura 17: $k = 10$

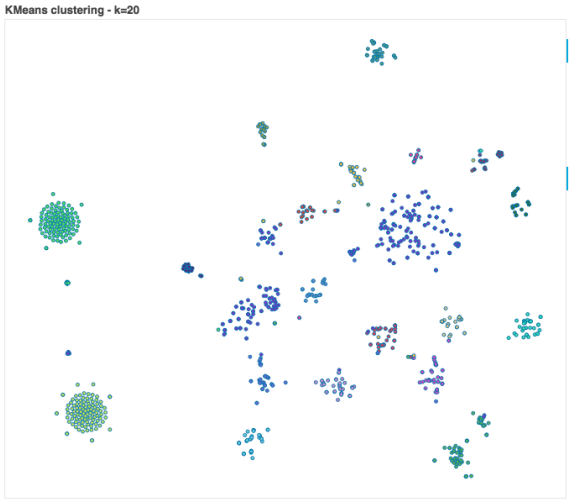

Figura 18: $k = 20$

Notamos que en esta base de datos, los clusters se ven mucho mas claros que en el dataset completo, por lo que esperaríamos que se pudiera extraer mas información particular de cada cluster y tener un mejor clasificador.
Una vez tomado el subconjunto, entrenamos un Random Forest separando de igual manera un conjunto de Test y uno de Train y escogiendo los hiperparámetros usando *GridSearchCV*, de manera que los mejores fueron $max\_features = auto, n\_estimators = 100, max\_leaf\_nodes = 13$.

A continuación presentamos los resultados de algunas métricas.

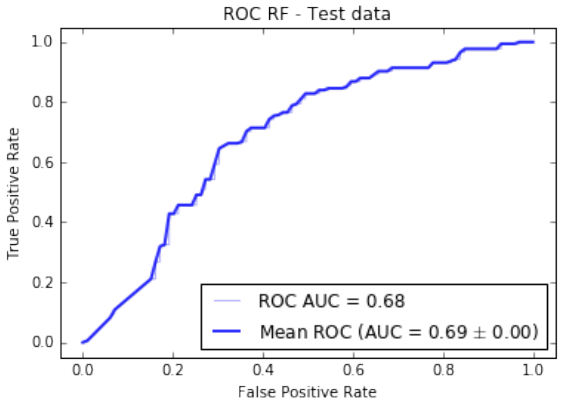

Figura 19: Curva ROC

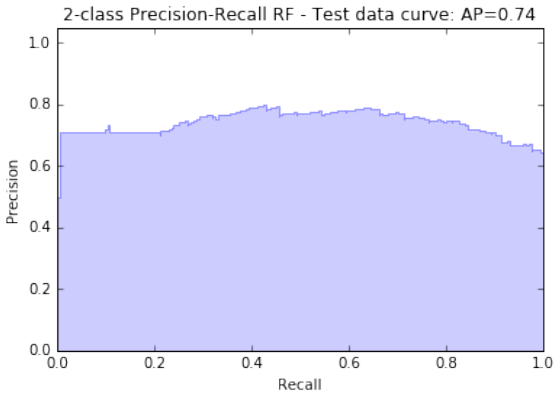

Figura 20: Precision-Recall

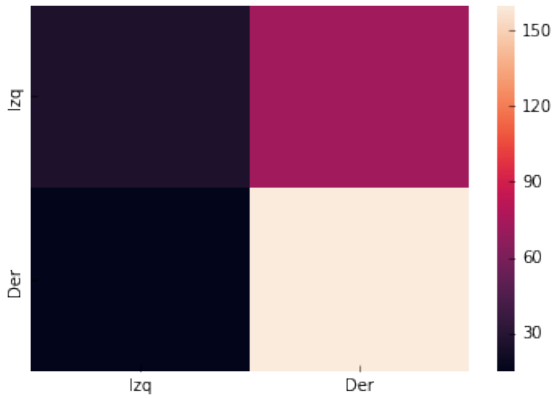

Figura 21: Matriz de confusión

Notamos que nuestro supuesto no parece ser cierto, al menos en el sentido de la información que nos proveetf-idf. Intentamos ahora sin usar la reducción de dimensión *SVD*, para ver si los resultados mejoran. A continuación presentamos los resultados

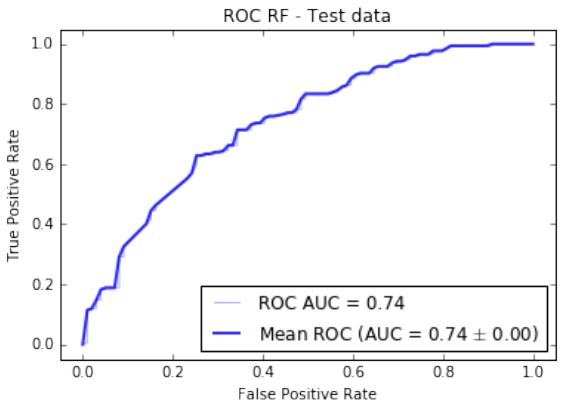

Figura 22: Curva ROC

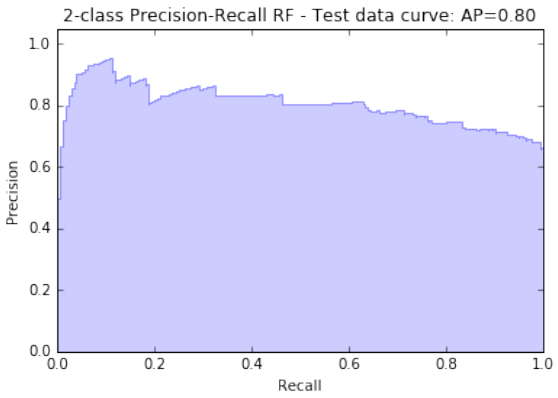

Figura 23: Precision-Recall

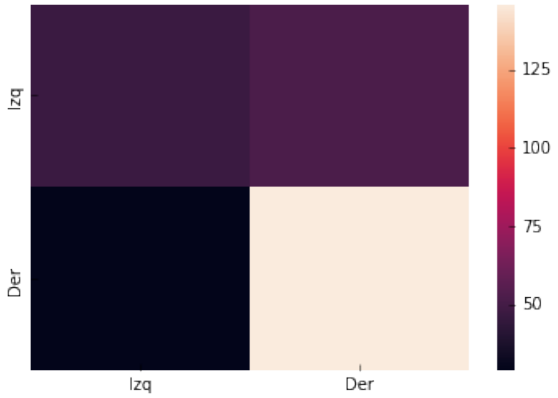

Figura 24: Matriz de confusión

Notamos una ligera mejoría que al usar la descomposición SVD en este subconjunto de la base de datos, sin embargo parece que el modelo se está sesgando por el hecho de haber mas noticias de "derecha" que de "izquierda", por lo que no creemos que pueda ser usado en un contexto de noticias generales, e incluso restringido a política.

6. Conclusiones

Dados los resultados, podemos ver que tf-idf es una buena técnica para hacer clustering de encabezados pues combinándola con LSE y K-Means pudimos ver que se agrupan bien los temas en los clusters identificados, con mejores resultados cuando aumentábamos el número de clusters, lo cual hace sentido ya que hay muchos temas y eventos que se cubren en las noticias diariamente, por lo que un número óptimo de clusters podría ser el número de temas y eventos que uno estime que han ocurrido en el periodo de tiempo transcurrido el lapso de tiempo, para el caso de las noticias.

En la parte de clasificación, sabíamos que era un problema muy complejo e incluso difícil de definir, sin embargo creemos que es importante abordarlo dada la relevancia que tendrá en nuestro país en los siguientes meses. Lo que podemos concluir es que usando la información de tf-idf y el cluster el que le fue asignado usando k-means, no es suficiente para determinar si un medio tiene cierta inclinación política.

A pesar de no haber encontrado un resultado conclusivo en la predicción, notamos que el análisis semántico (en el sentido de $tfidf$) es suficiente para notar la correlación entre la fuente que publica un medio y su inclinación política, creemos que hay muchos detalles por trabajar en esta área y que de ser trabajados, sería posible mejorar en gran medida nuestros resultados. Una posible siguiente linea de trabajo sería trabajar en una manera de complementar el enfoque con la extracción información gramática de los encabezados, o hacer un análisis del cuerpo entero de las noticias seleccionadas.

Referencias

[1] Lexiconista *Lemmatization Lists* http://www.lexiconista.com/datasets/lemmatization/, consultado el 13 de diciembre de 2017.

[2] Stanford NLP *Term frequency and weighting* https://nlp.stanford.edu/IR-book/html/htmledition/term-frequency-and-weighting-1.html, consultado el 13 de diciembre de 2017.

[3] Scikit Learn *Working With Text Data* http://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html#tokenizing-text-with-scikit-learn, consultado el 13 de diciembre de 2017.

[4] Sholar, M., Glaser N. *Predicting Media Bias in Online News*. Stanford, 2016.

[5] Alexa *Medios mas visitados en México* https://www.alexa.com/topsites/category/World/Espa%C3%B1ol/Regional/Am%C3%A9rica/M%C3%A9xico/Noticias_y_medios, consultado el 13 de diciembre de 2017.

[6] Ali, O., Flaounas,I. *Automating News Content Analysis: An Application to Gender Bias and Readability. In JMLR: Workshop and Conference Proceedings* 11, pp. 36-43.

[7] Mladenic, Dunja. *Learning How to Detect News Bias.* http://kt.ijs.si/markodebeljak/Lectures/Seminar_MPS/2012_on/Seminars_2014_15/Jenya%20Belyaeva/seminarIBelyaeva.pdf, consultado el 13 de diciembre de 2017.