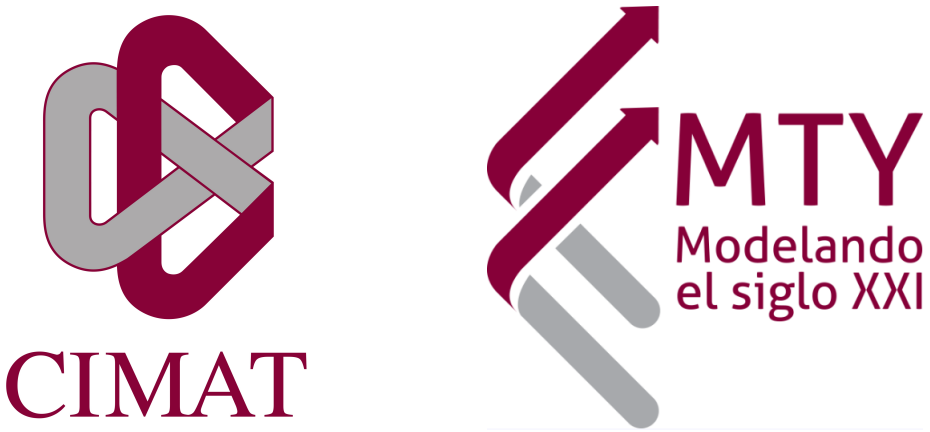
Tarea 3

Centro de Investigación en Matemáticas



Maestría en Cómputo Estadístico
Ciencia de Datos

Isaias Siliceo Guzmán

22 de abril de 2024

1. Problema 1

Considera los datos `MNIST` de dígitos escritos a mano que usamos anteriormente de 28×28 píxeles. Para mayor facilidad, puse los datos en archivos `csv` (`mnist.zip`):`mnist_Xtrain.csv`, `mnist_Ytrain.csv` contienen los valores de los píxeles (normalizados) y su respectiva categoría para entrenamiento, y (`mnist_Xtest.csv`, `mnist_Ytest.csv`), lo mismo para los datos de prueba. La figura 1.1 muestra un ejemplo de éstos datos, el cual se generó con el Código MNIST.

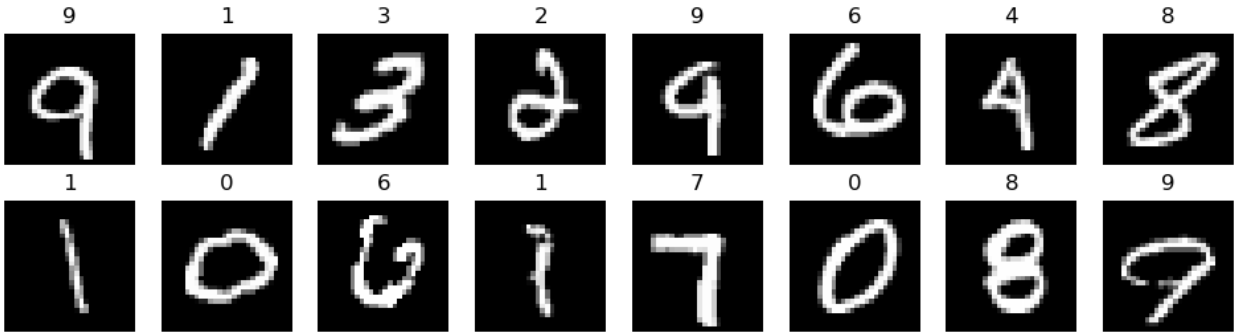


Figura 1.1: Ejemplo de los dígitos MNIST, donde se muestra también, la etiqueta correspondiente.

```
import pandas as pd
import numpy as np

train_x = pd.read_csv("mnist_Xtrain.csv",header=None).to_numpy()
train_y = pd.read_csv("mnist_Ytrain.csv",header=None).to_numpy()

figure = plt.figure(figsize=(12, 3))
cols, rows = 8, 2
for i in range(1, cols * rows + 1):
    sample_idx = np.random.choice(len(train_y), 1)[0]
    img, label = train_x[sample_idx].reshape((28,28)), train_y.squeeze()[sample_idx]
    figure.add_subplot(rows, cols, i)
    plt.title(label)
    plt.axis("off")
    plt.imshow(img.squeeze(), cmap="gray")
```

Figura 1.2: Código MNIST: Lectura y visualización de los datos.

En este ejercicio implementarás métodos de clasificación para los $k \in K = \{0, 1, \ldots, 9\}$ dígitos.

a) Implementa el *baseline* que usaremos. Este será un método de regresión multivariada, es decir

$$\mathbf{Y} = \mathbf{X}\hat{\mathbf{B}},$$

donde $\mathbf{Y}_{n\times|K|}$ es una matriz indicadora, donde cada renglón tiene ceros excepto en el lugar que corresponde al valor y_k, donde colocamos un 1. Por ejemplo, si alguna imagen corresponde al dígito "3", el renglón correspondiente en $\mathbf{Y}$ será $(0, 0, 0, 1, 0, 0, 0, 0, 0, 0)$.

$\mathbf{X_{n\times 748}}$ es la matriz de características y $\hat{\mathbf{B}}$ es la matriz cuyas columnas contienen los

$|K|$ coeficientes correspondientes $\hat{\beta}_k$.

Con esta formulación, asumimos un modelo lineal para cada respuesta y_k:

$$\hat{y}_k = \mathbf{X}\hat{\beta}_{\mathbf{k}},$$

y la clasificación para alguna observación x se obtiene mediante

$$\hat{C}(x) = \arg\max_{k \in K} \hat{y}_k.$$

Utiliza las tuplas `(x_train, y_train), (x_test, y_test)` que usamos en clase para ajustar y probar el modelo, respectivamente. Puedes restringir el número de observaciones de cada conjunto, pero procura que el conjunto de entrenamiento sea más grande que el de prueba. Reporta las métricas de evaluación del clasificador.

b) Utiliza clasificadores basados en LDA y QDA. Verifica si puedes superar al baseline respecto a las métricas que obtuviste. ¿Crees que ayudaría tener otra representación de los dígitos? Explica tu respuesta e impleméntala.

Las métricas de evaluación resumidas para el inciso (a) y (b) se presentan en la tabla 1.1. Estas métricas se obtuvieron de la aplicación de una regresión lineal multivariada (*baseline*), un Análisis Discriminante Lineal (LDA) y un Análisis Discriminante Cuadrático (QDA) sobre los datos de entrenamiento y prueba tal como se encuentran en MNIST.zip $(60k, 10k)$. Observándose que el *baseline* tiene un buen desempeño para hacer predicciones, con un 86 % de exactitud sobre los datos de prueba; justo por debajo del método LDA, el cual tiene una exactitud en la predicción de 87 %. Finalmente, se probó el modelo de QDA del mismo modo y este mostró estar carecer de exactitud sin haber hecho algún preprocesamiento de los datos originales.

Métrica	*baseline*	LDA	QDA	PCA-LDA	PCA-QDA	FA-LDA	FA-QDA
Precisión	.86	.87	.72	.87	.86	.87	.94
Recall	.86	.87	.55	.87	.81	.87	.94
f1-score	.86	.87	.50	.87	.82	.87	.94

Cuadro 1.1: Métricas obtenidas en el análisis de la base de datos MINIST.zip. Para generar los resultados de las primeras tres columnas (*baseline,LDA,QDA*) los datos de entrenamiento y prueba se tomaron tal y como se encontraron en la proporción $(60k, 10k)$. Para las columnas correspondientes a (PCA-LDA,PCA-QDA,FA-LDA,FA-QDA) se utilizó una proporción diferente $(45k, 15k)$ para evitar un sesgo.

Obtener otra representación de los datos en baja dimensión podría ser de ayuda para hacer una predicción más certera basada en las características más relevantes de los datos. En particular, se eligió el Análisis de Componentes Principales, por ser un método ampliamente utilizado para hacer reducción de dimensiones y por otro lado, se eligió el Método de Análisis de Factores, el cual puede verse como una generalización del método de PCA. Para hacer la reducción de dimensiones es necesario tener los datos completos y no es viable unirlos bajo algún criterio como concatenación porque esto podría causar un sesgo; de modo que, se utilizaron los datos de entrenamiento `train_x` y `train_y` para hacer una nueva división de los datos apropiada para hacer una reducción de dimensiones. Esta nueva proporción quedó en $(45k, 15k)$.

1.1. Análisis de Componentes Principales (PCA)

Para el análisis de PCA se eligieron 256 componentes para realizar la reducción de dimensiones. Tomando en cuenta que las dimensiones originales son $28 \times 28 = 784$, es una reducción considerable. Sin embargo, se puede observar en la figura 1.3 la varianza acumulada en función del número de componentes, la cual indica claramente que las primeras 100 componentes bastan para explicar el 70 % de la varianza acumulada de los datos originales. Por lo cual, se utilizan sólo las primeras 100 para hacer los análisis discriminantes.

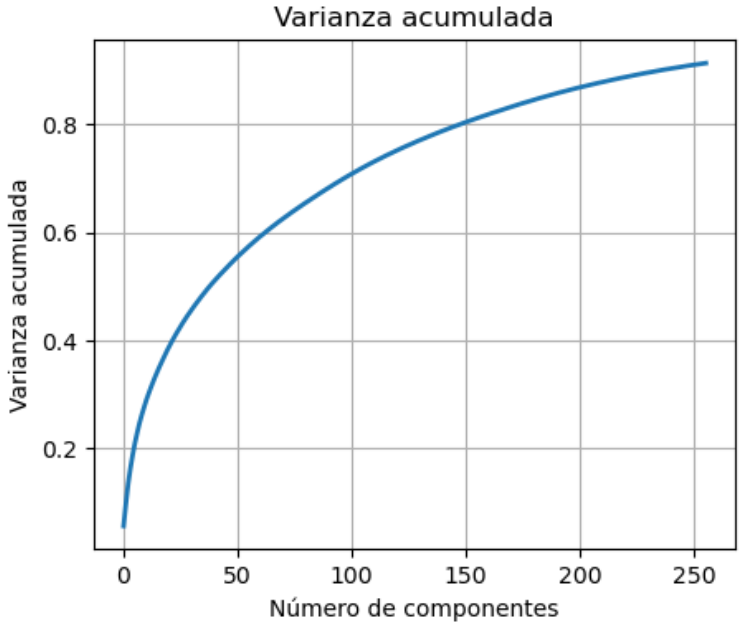


Figura 1.3: Varianza acumulada al aplicar PCA en el preprocesamiento de los datos `x_train`. Se observa que las primeras 100 componentes son suficientes para recuperar el 70 % de la varianza acumulada de los datos originales.

1.1.1. PCA-LDA

Para aplicar el Análisis Discriminante Lineal con PCA se utilizaron las primeras 100 componentes de la reducción anterior. Aplicar esta reducción como preprocesamiento a los datos originales no muestra una mejora en las métricas del Cuadro 1.1, se mantienen igual.

1.1.2. PCA-QDA

Por otro lado, se aplicó un Análisis Disriminante Cuadrátrico con un preprocesamiento basado en Análisis de Componentes Principales (utilizando las mismas 100 primeras componentes) dando como resultado una mejora muy relevante en las métricas de precisión, recall y f1-score, alcanzando 86 %, 81 % y 82 % respectivamente.

1.2. Análisis de Factores (FA)

El Análisis de Factores es un método de reducción de dimensión que también busca dar una interpretación a los factores en los que se reduce. En este caso, no se buscó una interpretación en

particular, únicamente una reducción drástica de los datos que conservara una buena cantidad de la varianza original. Para ello se utilizaron 64 componentes para generar una matriz de puntuaciones apropiada para hacer nuevamente los análisis discriminantes.

1.2.1. FA-LDA

Al hacer el análisis con la matriz de puntuaciones se observó algo similar a lo que ocurrió con el Análisis por Componentes Principales. Las métricas se mantuvieron en 87 % tal como en los incisos anteriores.

1.2.2. FA-QDA

Este análisis arrojó los mejores resultados, dando una exactitud del 94 %. Siendo una mejora radical con respecto a los datos originales sin preprocesamiento.

c) **Opcional (puntos extra)**. Programa una aplicación interactiva donde dibujes un número y te diga qué dígito es usando los clasificadores del inciso anterior. Puedes usar y modificar el applet que usamos en el curso.

A continuación en las figuras 1.4, 1.5, 1.6, 1.7 y 1.8 se muestran algunas imágenes resultado de la implementación del un applet que hace una predicción utilizando los modelos ajustados de LDA y QDA. Aunque se probaron aquellos cuyo preprocesamiento dio una mejora en los resultados de exactitud, al hacer las predicciones con un número dibujado por el usuario, no muestran buenos resultados. Esto podría indicar que aquellos modelos con reducción de dimensiones podrían estar sobreajustados o también podría deberse al cambio de dimensiones del canvas a una matriz que se pueda analizar con el modelo apropiado.

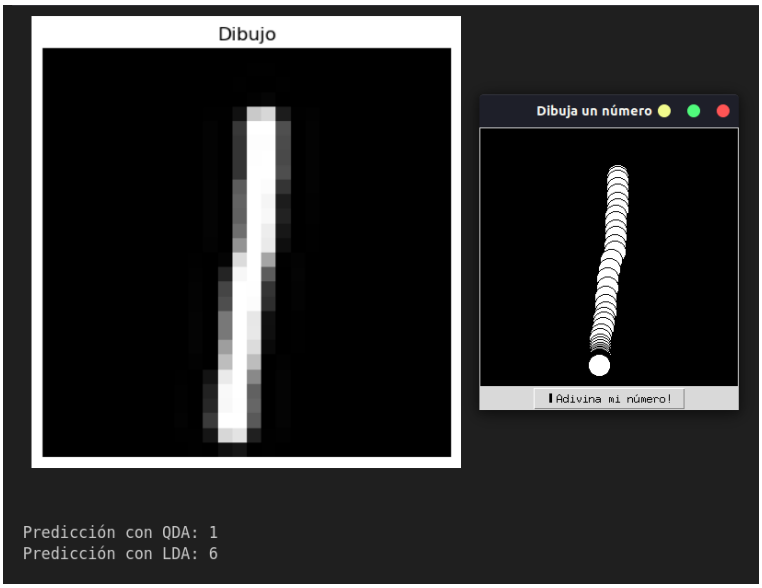


Figura 1.4: Se dibuja un número 1. Se observa el resultado de aplicar QDA y LDA sin preprocesamiento. Con QDA muestra un resultado correcto mientras que con LDA un resultado erróneo.

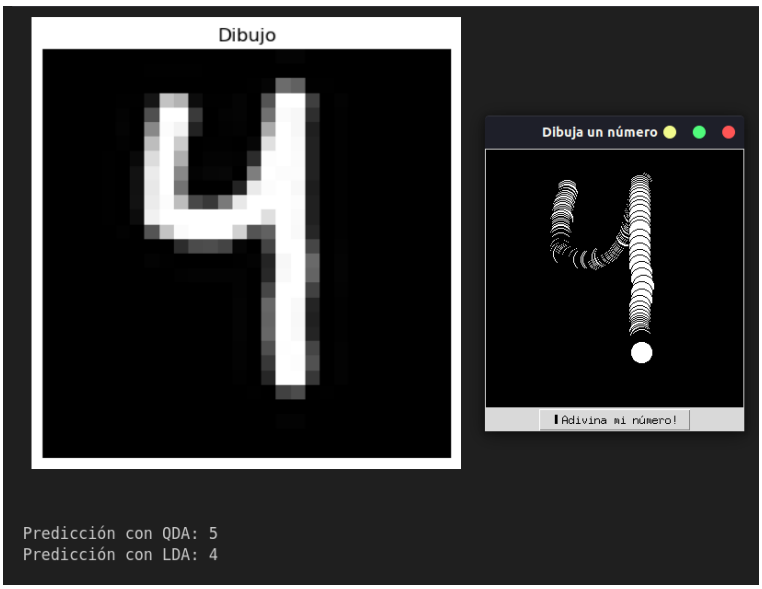


Figura 1.5: Se dibuja un número 4. Se observa el resultado de aplicar QDA y LDA sin preprocesamiento. Con QDA muestra un resultado erróneo mientras que con LDA un resultado correcto.

Figura 1.6: Se dibuja un número 5. Se observa el resultado de aplicar QDA y LDA sin preprocesamiento. Con QDA muestra un resultado correcto mientras que con LDA un resultado erróneo.

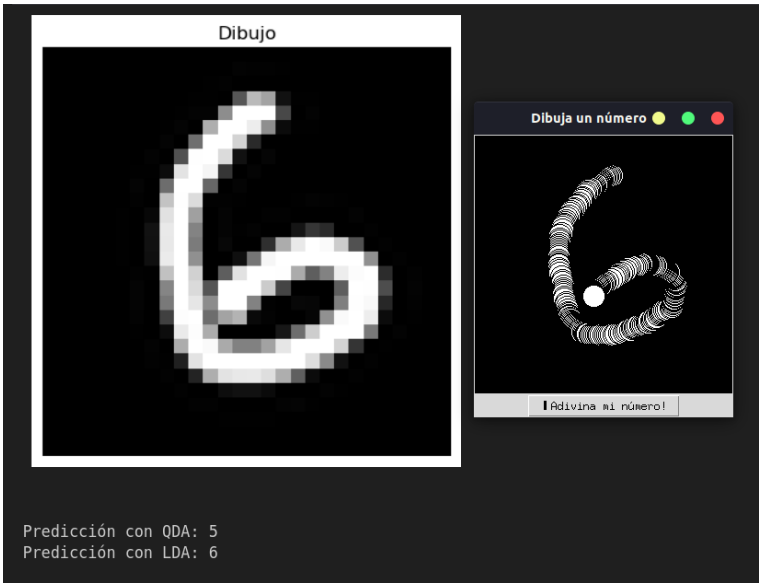


Figura 1.7: Se dibuja un número 6. Se observa el resultado de aplicar QDA y LDA sin preprocesamiento. Con QDA muestra un resultado erróneo mientras que con LDA un resultado correcto.

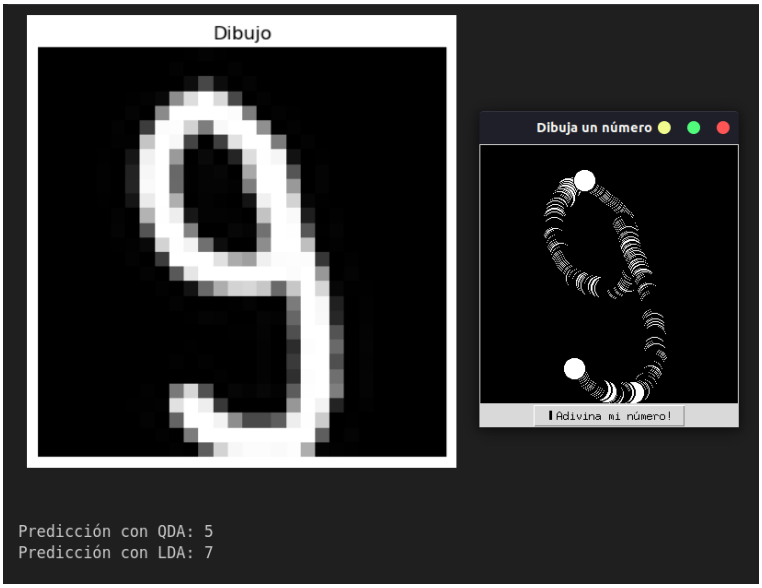


Figura 1.8: Se dibuja un número 9. Se observa el resultado de aplicar QDA y LDA sin preprocesamiento. Con QDA muestra un resultado erróneo y con LDA un resultado erróneo.

2. Problema 2

Este ejercicio es sobre análisis de tópicos.

Un tópico es una variable latente que representa o resume conceptos importantes de un texto, como el significado o las ideas principales del mismo. Un tópico, se conforma por varias palabras relacionadas semánticamente entre si de acuerdo a cierto contexto. En el área de procesamiento de lenguaje natural (NLP), forma parte de una tarea general llamada *recuperación de información* (IR). Para nosotros, desde la perspectiva de machine learning, la consideraremos como una tarea de aprendizaje no-supervisado a partir de una representación vectorial particular de los textos.

Considera una representación documento-término como las que vimos en clase. Una forma sencilla de extraer estructruras latentes entre documentos y términos es usando análisis semántico latente (LSA), el cual se basa en factorizaciones apropiadas de ésa matriz. Sea $\mathbf{A}_{m\times n}$ la matriz TF-IDF de rango r, con m renglones (documentos) y n columnas (términos). Una aproximación de rango k de ésta matriz, está dada por la factorización SVD $\mathbf{A} \approx \mathbf{A}^{(k)} = \mathbf{U}^{(k)}\mathbf{\Sigma}^{(k)}\mathbf{V}^{(k)\prime}$, donde $\mathbf{\Sigma}^{(k)}$ es diagonal[a] con los k eigenvalores más grandes de $\mathbf{A}$ y $\mathbf{U}^{(k)}$, $\mathbf{V}^{(k)}$ contienen los correspondientes eigenvectores izquierdos y derechos que definen una base ortonormal para los espacios columna y renglón, respectivamente. Al aplicar ésta factorización en matrices documento-término, podemos extraer las relaciones semánticas y conceptuales entre documentos y términos expresadas en un conjunto de componentes (o tópicos) k, mediante representaciones densas y de baja dimensión, donde $\mathbf{V}^{(k)}_{n\times k}$ y $\mathbf{U}^{(k)}_{m\times k}$ nos proporcionan una representación de los términos y documentos, respectivamente en términos de los k tópicos, y $\mathbf{\Sigma}^{(k)}$ nos proporcionan la *importancia* de cada tópico. En `python`, puedes usar la implementación de `sklearn.decomposition.TruncatedSVD`.

En éste ejercicio, realizarás un análisis de tópicos en las transcripciones de las conferencias matutinas de la presidencia de México, a los cuales puedes acceder en éste repositorio. [b] Para construir tu modelo de tópicos, considera los textos de las conferencias *por semana* durante los años 2019 a 2023, usando las transcripciones que corresponden al presidente, contenido en los archivos *"PRESIDENTE ANDRES MANUEL LOPEZ OBRADOR.csv"*[c].

a) Obtén una representación TF-IDF de los textos. Define el tamaño del vocabulario y realiza el preproceso que consideres necesario en los textos, considerando que para un análisis de tópicos, no es recomendable que el vocabulario sea tan grande, y es mejor conservar palabras cuyo uso dentro del texto, pueda asociarse con tópicos, Documenta y justifica tus parametrizaciones.

[a] Estamos considerando la representación *reducida* de SVD, donde se han removido todas las entradas cero de $\mathbf{\Sigma}$, y las correspondientes columnas de $\mathbf{U}$ y $\mathbf{V}$, y aún más, se han reemplazado por ceros los $k - r$ valores propios más pequeños.

[b] Recomiendo hacer un git clone a todo el repositorio, para mantener la misma estructura de archivos.

[c] Ten cuidado con las diferentes variaciones del nombre de los archivos, e,g: LOPEZ y LÓPEZ...

Para la lectura de estos archivos se realizó una copia total del repositorio en cuestión y luego se aplicó una búsqueda recursiva en las carpetas para recolectar la ruta completo de todos los archivos que contuvieran la terminación "OR.csv". Los datos se introdujeron a un DataFrame apropiadamente y se filtraron entre las fechas 2019-01-01 y 2023-12-31. Finalmente. Se agruparon los textos por semana

para dar un total de 261 semanas a lo largo de los 5 años de estudio. Por otro lado, Utilizando el la clase preprocesaTexto proporcionada en clase, se remueven acentos y palabras de stop. En la figura 2.1 se muestra la bolsa de palabras correspondiente al texto después de una primera limpieza de palabras. Se procedió a hacer una segunda limpieza basada en remover las palabras más repetidas de esta bolsa que no tienen relevancia o no pueden ser asociados a un tópico en particular. Luego

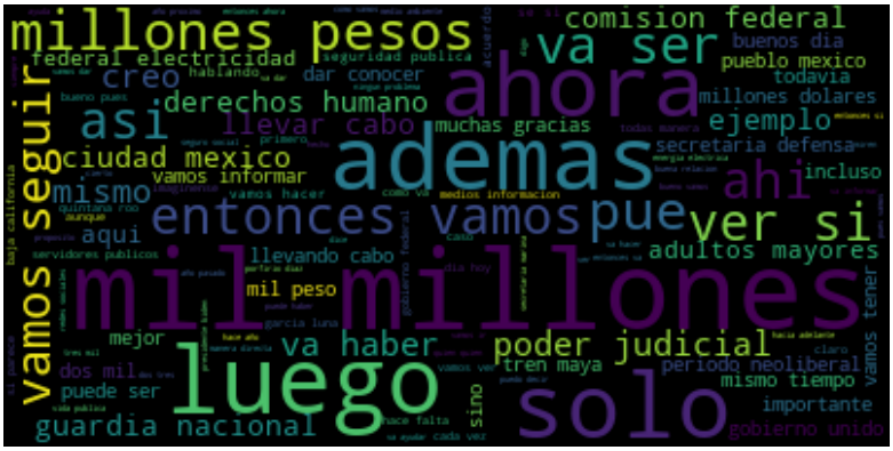


Figura 2.1: Bolsa de palabras previa a la segunda limpieza de palabras. Esta se graficó con el objetivo de identificar las palabras que más se repiten más y no están asociadas a un tópico particular.

de la segunda limpieza de palabras se procedió a realizar una nueva bolsa de palabras para observar si las más relevantes pueden ahora ser asociadas a un tópico en particular, dando como resultado la bolsa mostrada en la figura 2.2

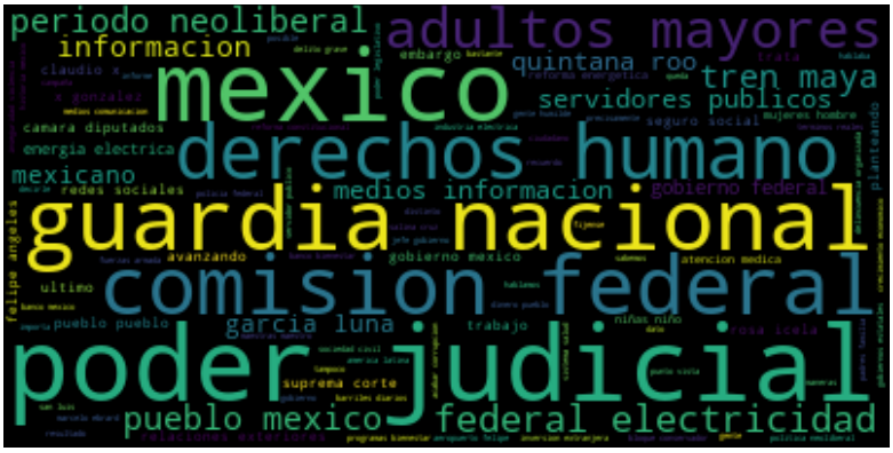


Figura 2.2: En esta bolsa de palabras se puede observar que las palabras más relevantes pueden ser asociadas a un tópico particular o que son palabras relevantes para un estudio de tópicos.

Para este estudio se tomó un máximo de 100 palabras, ya que se observó que las primeras 100 más repetidas podían ser adecuadamente asociadas a un tópico en particular. En la figura 2.3 se muestra la representación TF-IDF, la cual tiene 261 registros y 100 palabras por cada registro.

```
9  tf_idf_df
✓ 7.6s
```

	adultos	aeropuerto	agua	atencion	autoridades	avanzando	bienestar
0	0.014311	0.000000	0.000000	0.051820	0.053428	0.026309	0.012906
1	0.050559	0.024515	0.000000	0.015256	0.047187	0.069707	0.053191
2	0.046820	0.015134	0.000000	0.000000	0.014566	0.078896	0.084441
3	0.055705	0.006752	0.000000	0.037819	0.032494	0.038401	0.163258
4	0.095735	0.066313	0.006631	0.030950	0.012764	0.050283	0.117164
...	...	...	...	...	...	...	...
256	0.027503	0.111126	0.062231	0.033195	0.064171	0.088477	0.057869
257	0.050829	0.028166	0.091540	0.105169	0.027108	0.120136	0.052384
258	0.000000	0.033948	0.064501	0.079223	0.042474	0.070789	0.072607
259	0.008573	0.000000	0.108079	0.023282	0.032006	0.086681	0.069579
260	0.019108	0.172947	0.018530	0.046126	0.005945	0.011709	0.034463

261 rows × 100 columns

Figura 2.3: Representación TF-IDF

b) Obtén k tópicos mediante la descomposición SVD. Elige un k adecuado y justifícalo. Representa cada tópico mediante un `word cloud`[a] de los términos que forman cada tópico según la importancia expresada en las magnitudes de los renglones de $\mathbf{V}^{(k)}$. ¿Puedes asignar un "nombre" representativo a cada tópico?

[a] Puedes usar el módulo wordcloud de python, el cual tiene bastantes ejemplos, incluyendo la opción `generate_from_frequencies`, que puede ser de utilidad.

De acuerdo con las palabras con las que se construyó la representación TF-IDF, se lograron identificar al menos 5 categorías relevantes para el estudio; Seguridad y Violencia, Sector Privado, Salud, Desarrollo y Economía. Sin embargo, estos tópicos se ajustaron de acuerdo a las bolsas de palabras que se muestran en las figuras 2.4, 2.5, 2.6, 2.7 y 2.8. Estas se asocian a una bolsa en particular de acuerdo con los pesos de la matriz $V^{(k)}$.

Figura 2.4: Esta bolsa de palabras se puede asociar al tópico "Nación y Desarrollo".

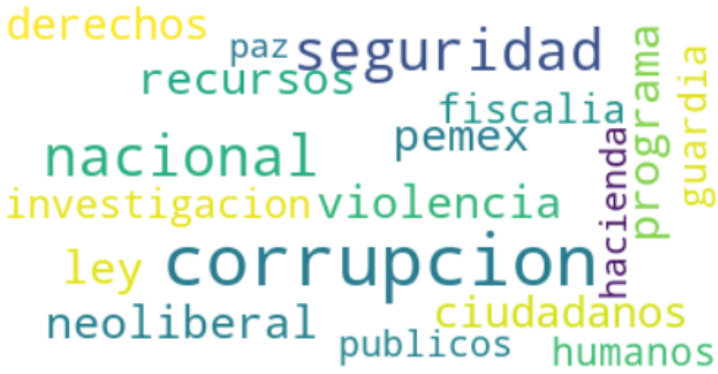


Figura 2.5: Esta bolsa de palabras se puede asociar al tópico "Corrupción, Seguridad y Violencia".

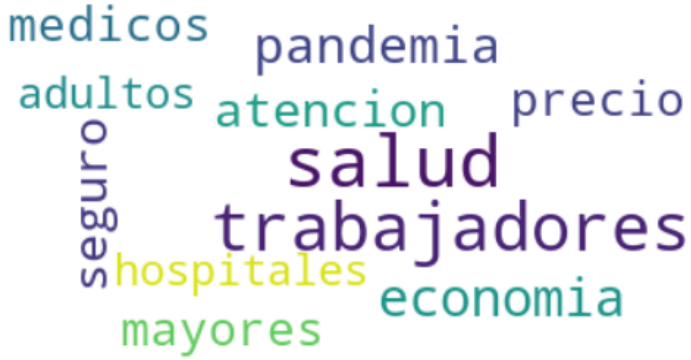


Figura 2.6: Esta bolsa de palabras se puede asociar al tópico "Salud Pública".

Figura 2.7: Esta bolsa de palabras se puede asociar al tópico "Sector Privado".

Figura 2.8: Esta bolsa de palabras se puede asociar al tópico "Economía y Relaciones Exteriores".

c) Usando el modelo de tópicos ajustado en el paso previo, obtén la representación correspondiente de *cada una* de las conferencias del presidente durante los años del estudio, calculando la matriz documento-tópico mediante el producto $\mathbf{XV}^{(k)}$ (o con el método `transform` de `TruncatedSVD`). Asigna cada conferencia a su tópico correspondiente usando como criterio el valor máximo de cada renglón de la matriz. Usa visualizaciones de baja dimensión basadas en PCA, Kernel PCA y t-SNE de la asignación de tópicos que obtuviste. ¿Observas patrones interesantes? Describe brevemente tus hallazgos.

Al calcular la matriz documento-tópico mediante la descomposición SVD, es evidente que los pesos se aglomeran en la primera columna, lo cual se traduce en que no se haga una buena predicción de a qué tópico corresponde cada registro. De hecho, todos los registros se agrupan en un sólo tópico, lo cual es incorrecto. En las figuras 2.9 y 2.10 se muestran los resultado utilizando la matriz dada por el producto $XV^{(k)}$.

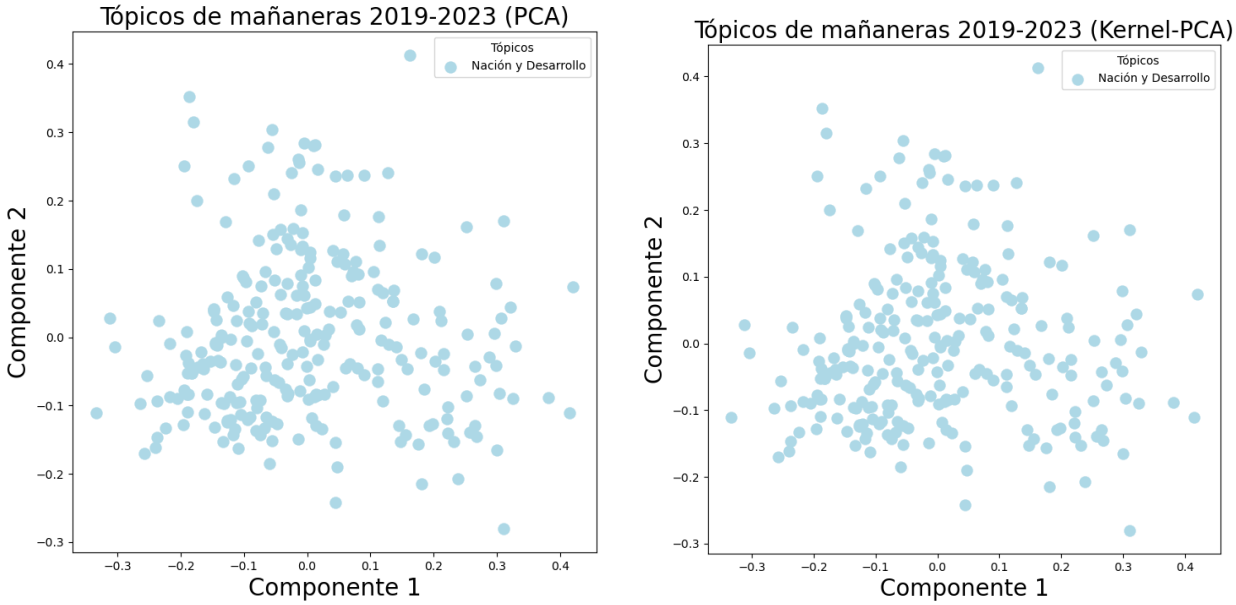


Figura 2.9: Se calculó el argumento máximo de cada registro en la matriz documento tópico calculada mediante el producto $XV^{(k)}$.

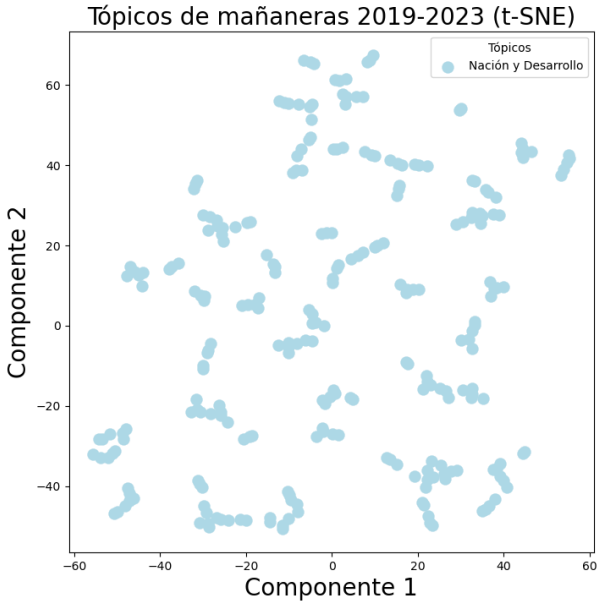


Figura 2.10: Tópicos calculados mediante el argumento máximo de cada registro en la matriz documento-tópico generada mediante el producto $XV^{(k)}$. Las entradas de esta matriz en la primera columna son muy grandes con respecto al resto de columnas. Teniendo incluso algunos casos con muchos registros negativos.

d) Un problema que surge al usar SVD es la falta de interpretabilidad, ya que no es claro cómo pueden considerarse los valores negativos en las matrices $\mathbf{U}$ y $\mathbf{V}$. Una forma de resolver éste problema es usar una factorización no-negativa de matrices (NMF), que es adecuada para matrices con entradas no negativas, como las TF-IDF. Para una matriz $\mathbf{A}$ de rango r con entradas no-negativas, NMF calcula una aproximación de rango $k < r$ mediante la factorización $\mathbf{A} \approx \mathbf{A}^{(k)} = \mathbf{W}^{(k)}\mathbf{H}^{(k)}$, donde $\mathbf{W}^{(k)}, \mathbf{H}^{(k)} \geq 0$. En `scikit-learn` puedes usar la clase NMF del módulo `sklearn.decomposition.NMF`. Repite los incisos anteriores usando ésta descomposición. ¿Cuál te parece mejor y por qué?

En las figuras 2.11, 2.12, 2.13, 2.13, 2.14 y 2.15 se muestran los resultados para las bolsas de palabras generadas a partir del argumento máximo por registro en cada uno de los 261 renglones de la matriz H de la factorización NMF. Algunas categorías muestran ser diferentes por lo cual se decidió dividir el tópico de "Nación y Desarrollo" en dos tópicos y dejar fuera a "Relaciones Exteriores". La factorización NMF es una mejor opción para realizar este estudio ya que permite encontrar patrones de separación adecuados en baja dimensión. Como se observan en las figuras 2.16 y 2.17.

Figura 2.11: Esta bolsa de palabras se puede asociar al tópico "Nación".

Figura 2.12: Esta bolsa de palabras se puede asociar al tópico "Desarrollo".

Además, en las figuras

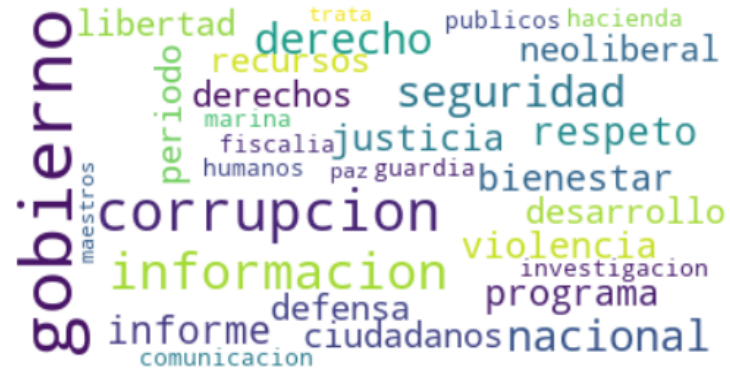


Figura 2.13: Esta bolsa de palabras se puede asociar al tópico ”Corrupción, Seguridad y Violencia”.

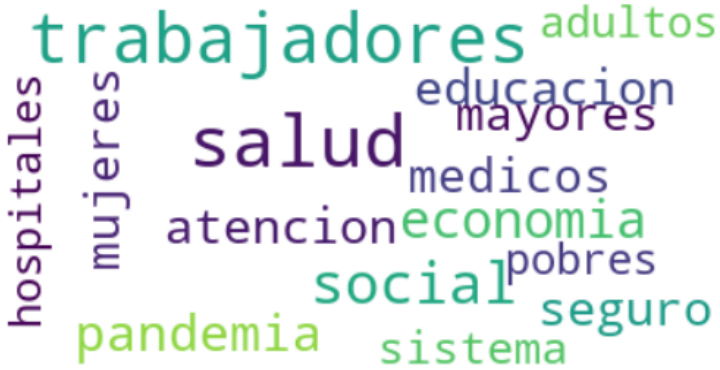


Figura 2.14: Esta bolsa de palabras se puede asociar al tópico ”Salud Pública”.

Figura 2.15: Esta bolsa de palabras se puede asociar al tópico ”Sector Privado”.

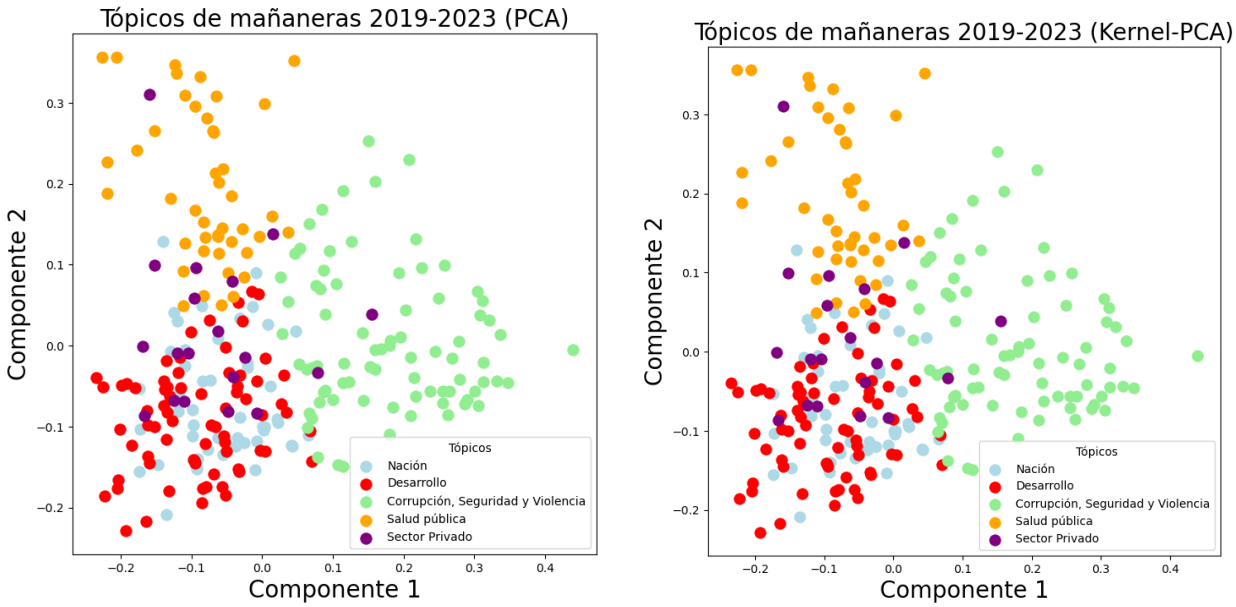


Figura 2.16: Se calculó el argumento máximo de cada registro en la matriz documento tópico calculada mediante el producto XH. Se pueden observar patrones claros de separación de los tópicos en ambas representaciones, siendo estas muy similares.

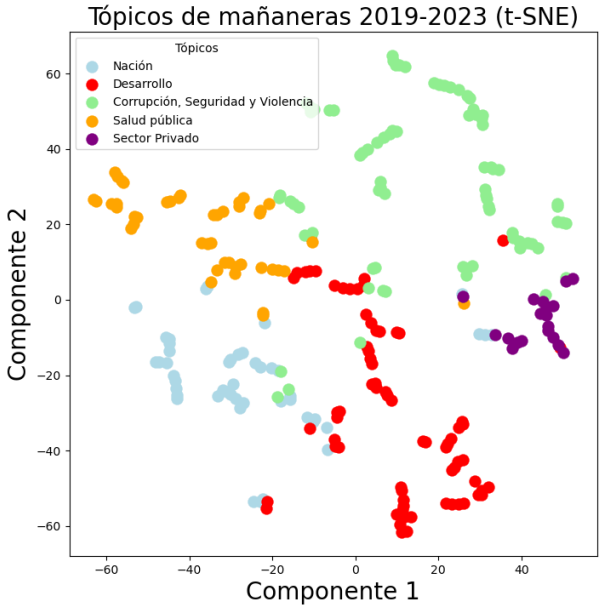


Figura 2.17: Tópicos calculados mediante el argumento máximo de cada registro en la matriz documento-tópico generada mediante el producto XH. Ya que esta matriz H no cuenta con elementos negativos, se muestra mejor distribuida que la matriz $XV^{(k)}$, de modo que se obtiene una buena separación por tópicos. Es posible observar patrones de separación en la proyección de los tópicos.

e) Usando los resultados del método que te parezca más conveniente, (SVD, NMF) construye un indicador semanal para cada uno de los k tópicos durante el periodo de estudio, basado en su frecuencia de aparición. Normalízalos de manera adecuada para que sean comparables y grafícalos como una serie de tiempo. Lo anterior, puede darte un panorama general de la dinámica de los temas que se han tratado en las conferencias matutinas. Realiza un reporte ejecutivo de tus análisis y hallazgos, resaltando las ventajas y desventajas de las metodologías exploradas y da tus conclusiones, incluyendo sugerencias para mejorar el análisis. [a]

[a] En el Moodle del curso, hay un par de artículos de referencia sobre LSA, que quizá puedan servir para ampliar algunos detalles del ejercicio, en caso de ser necesario. En éste ejercicio no está permitido usar módulos especializados en LSA, sólo aquellos que se mencionan en los incisos.

Finalmente, con la ayuda de la matriz Documento-Tópico determinada mediante el producto XH utilizada en el ejercicio anterior, se realizó la construcción de un indicador de la relevancia de cada tópico a lo largo de los 5 años de estudio. A continuación se resumen los resultados en las figuras 2.18, 2.19 y 2.20. Las tendencias y características de estos gráficos se muestran en sus descripciones. Así mismo, en el notebook anexo a este documento se encuentran algunas series de tiempo extras correspondientes a cada uno de los años del estudio, lo cual permite ver un poco más de cerca el comportamiento semanal de los tópicos a lo largo del tiempo.

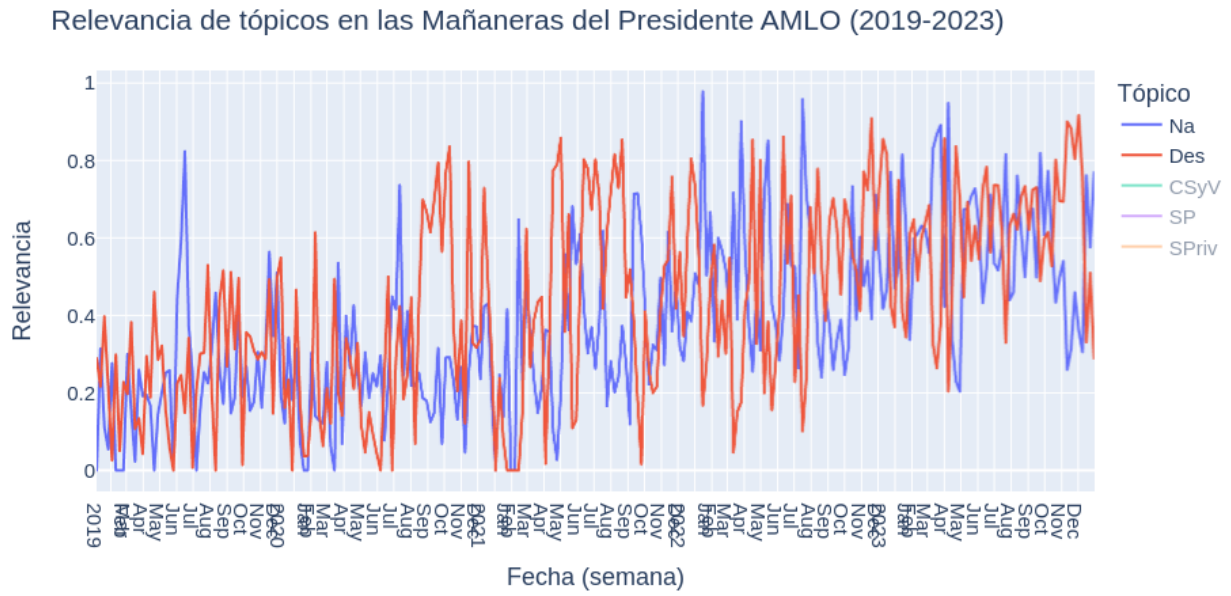


Figura 2.18: En esta serie de tiempo se encuentran los Tópicos "Nación" (Na) y "Desarrollo" (Des), los cuales han cobrado mayor relevancia en el paso de los 5 años de estudio.

A manera de conclusión, la factorización NMF es la más apropiada para realizar este estudio, ya que permite observar patrones relevantes en los datos y la matriz documento-tópico XH tiene los pesos mejor distribuidos en comparación a la $XV^{(k)}$. Además, en las series de tiempo es posible tener una perspectiva clara de la tendencia que ha habido a lo largo del tiempo, así como algunos sucesos muy relevantes. Como sugerencia, se propone analizar aquellos saltos más destacables para asociarlos con algún suceso importante en esas fechas, es probable que la razón por la cual se haya observado un pico en una fecha en particular se deba a un suceso relevante en México o en el mundo.

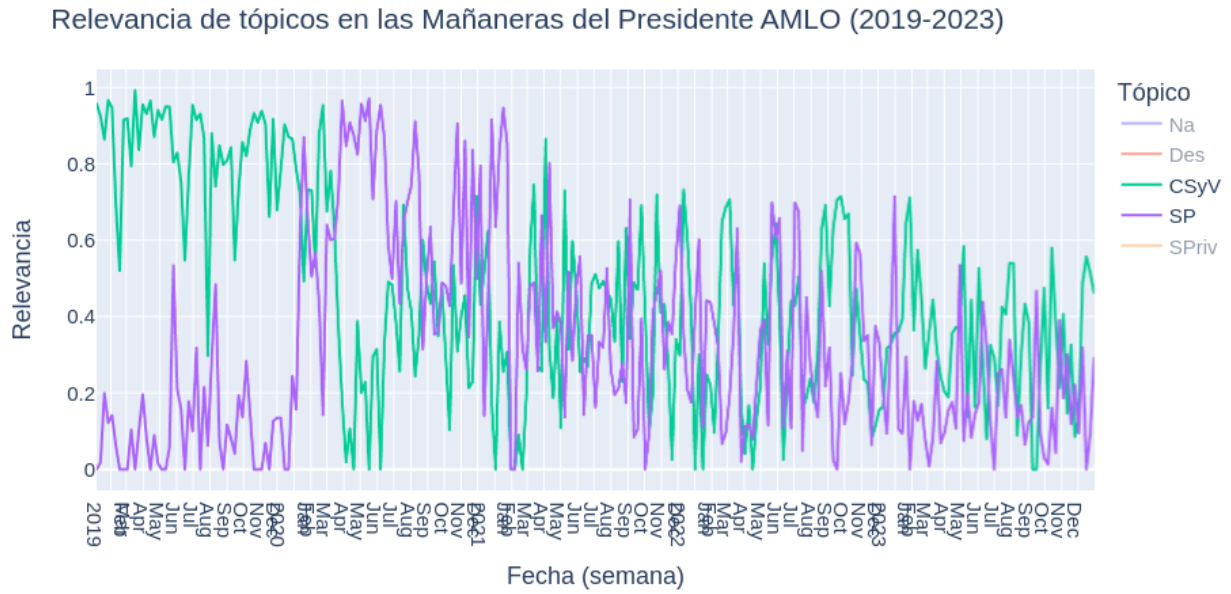


Figura 2.19: En esta serie de tiempo se encuentran los Tópicos "Corrupción, Seguridad y Violencia" (CSyV) y "Salud Pública" (SP), los cuales han perdido relevancia al paso de los 5 años de estudio. Al comienzo de 2019, los tópicos de CSyV son muy representativos Mientras que al inicio del año 2020 se ve un incremento abrupto en los temas de salud. Posiblemente, debido en gran medida al incio de la pandemia de 2020.

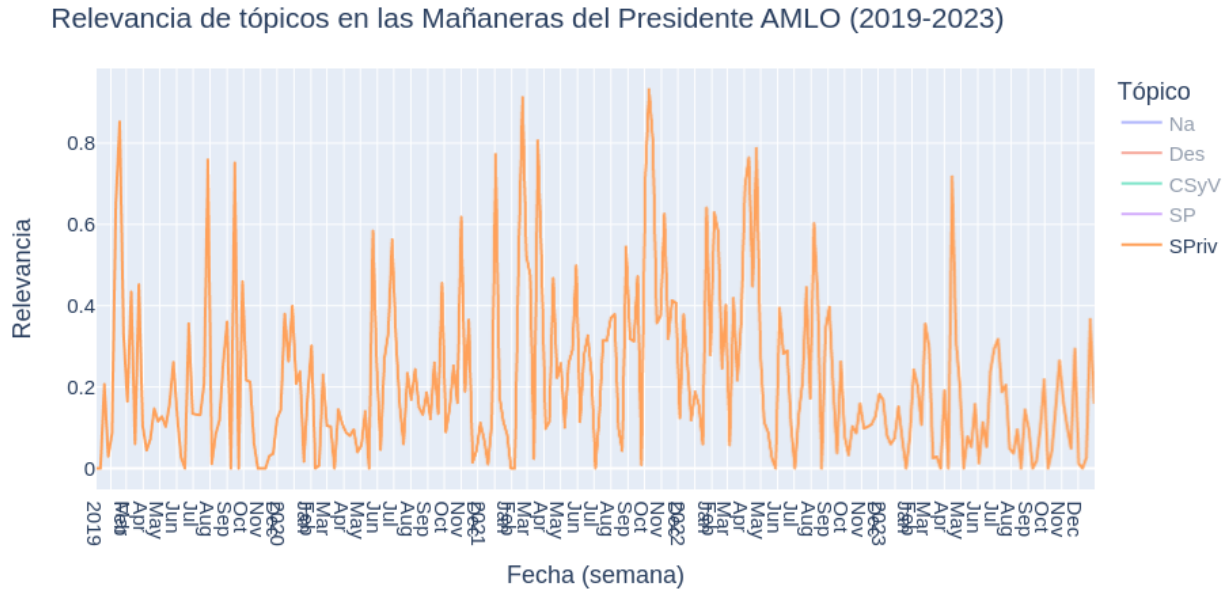


Figura 2.20: En esta serie de tiempo se encuentra el tópico "Sector privado" (SPriv). Este indicador no muestra una tendencia interesante a lo largo del tiempo. Pero con regularidad tiende a tener mucha relevancia y algunas semanas no. Esto puede deberse a que sea un tema recurrente pero ha mantenido un equilibro en la relevancia con respecto a otros tópicos.