

## Tarea 3

---

Luciano Andres Juárez López

22 de abril de 2024

### 1. PROBLEMA 1

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

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

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

donde  $Y_{n \times |K|}$  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  $Y$  será  $(0, 0, 0, 1, 0, 0, 0, 0, 0, 0)$ .  $X_{n \times 784}$  es la matriz de características, y  $\hat{B}$  es la matriz cuyas columnas contienen los  $|K|$  coeficientes correspondientes  $\hat{\beta}_k$ .

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

$$\hat{y}_k = X\hat{\beta}_k$$

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.
- 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.

### 1.1. SOLUCIÓN (PROBLEMA 1)

a)

Haciendo uso de los métodos proporcionados en "sklearn.metrics", obtendremos las métricas usando diferentes tamaños en nuestros conjuntos de muestra y entrenamiento para la clasificación:

Usando el 90 % de los datos en el conjunto de entrenamiento y de prueba, obtenemos los siguientes valores por las métricas utilizadas:

Número	Precision score
0	0.9020
1	0.8343
2	0.9142
3	0.8493
4	0.8138
5	0.8779
6	0.8780
7	0.8552
8	0.8417
9	0.8388

Cuadro 1.1: Podemos observar que los números que tuvieron un mejor puntaje en la predicción son 0 y 2

Observamos la matriz de confusión, donde podemos ver que el 1 número 1 tuvo el mayor número de aciertos y el 5 fue el que menos aciertos tuvo, de igual forma la mayoría ronda entre 700 a 900 aciertos.

$$\begin{bmatrix} 856 & 0 & 0 & 2 & 2 & 6 & 13 & 2 & 6 & 1 \\ 0 & 982 & 2 & 2 & 3 & 2 & 5 & 1 & 15 & 0 \\ 17 & 51 & 714 & 21 & 13 & 0 & 38 & 20 & 37 & 7 \\ 4 & 16 & 21 & 789 & 5 & 13 & 9 & 20 & 18 & 9 \\ 1 & 20 & 7 & 1 & 791 & 4 & 10 & 1 & 10 & 39 \\ 21 & 19 & 3 & 67 & 24 & 597 & 18 & 14 & 37 & 14 \\ 18 & 9 & 8 & 0 & 21 & 18 & 792 & 0 & 4 & 0 \\ 5 & 31 & 15 & 4 & 23 & 0 & 1 & 803 & 0 & 49 \\ 13 & 40 & 10 & 27 & 25 & 39 & 15 & 9 & 691 & 18 \\ 14 & 9 & 1 & 16 & 65 & 1 & 1 & 69 & 3 & 713 \end{bmatrix}$$

Usando el 70 % de los datos en el conjunto de entrenamiento y de prueba, obtenemos los siguientes valores por las métricas utilizadas:

Observamos la matriz de confusión, donde podemos ver que el 1 número 1 tuvo el mayor número de aciertos y el 5 fue el que menos aciertos tuvo, de igual forma la mayoría ronda entre 500 a 700 aciertos, mucho menor que cuando tomamos el 90 % de los datos del conjunto de entrenamiento y prueba.

Número	Precision Score
0	0.8996
1	0.8303
2	0.9065
3	0.8533
4	0.8029
5	0.8748
6	0.8703
7	0.8585
8	0.8433
9	0.8483

Cuadro 1.2: Podemos observar que el número que tuvo un mejor puntaje en la predicción fue el 2

$$\begin{bmatrix} 654 & 0 & 0 & 2 & 0 & 5 & 9 & 0 & 5 & 1 \\ 0 & 773 & 2 & 1 & 2 & 1 & 5 & 0 & 13 & 1 \\ 16 & 45 & 543 & 18 & 14 & 1 & 28 & 17 & 24 & 4 \\ 2 & 10 & 15 & 634 & 3 & 10 & 6 & 13 & 13 & 9 \\ 0 & 13 & 5 & 2 & 599 & 2 & 9 & 1 & 7 & 26 \\ 18 & 15 & 3 & 53 & 18 & 475 & 15 & 10 & 32 & 10 \\ 17 & 9 & 8 & 0 & 19 & 14 & 597 & 0 & 4 & 0 \\ 3 & 24 & 13 & 2 & 20 & 0 & 1 & 625 & 0 & 33 \\ 8 & 31 & 9 & 21 & 16 & 34 & 15 & 5 & 549 & 14 \\ 9 & 11 & 1 & 10 & 55 & 1 & 1 & 57 & 4 & 548 \end{bmatrix}$$

Usando el 60 % de los datos en el conjunto de entrenamiento y de prueba, obtenemos los siguientes valores por las metricas utilizadas:

Número	Score
0	0.8857
1	0.8363
2	0.9098
3	0.8593
4	0.8032
5	0.8766
6	0.8579
7	0.8513
8	0.8351
9	0.8505

Cuadro 1.3: Podemos observar que el número que tuvo un mejor puntaje en la predicción fue el 2, sin embargo la mayoria de los demas valores son menor a lo obtenido con el 70 % de los datos utilizados

En esta matriz de confusión podemos seguir observando que el número que mejor se clasifica es el 1 y conforme van disminuyendo el conjunto de datos, se va disminuyendo el número de casos que se clasifica

mal.

558	0	0	2	1	2	11	0	4	1
0	669	2	1	1	1	5	1	10	0
15	35	464	12	12	0	26	15	21	4
2	9	13	562	3	7	5	14	12	9
0	11	5	0	498	2	5	1	7	23
19	9	2	43	13	412	15	8	30	8
14	8	7	0	14	13	507	0	4	0
3	21	10	2	16	1	2	521	1	26
8	30	7	22	14	31	14	5	466	12
11	8	0	10	48	1	1	47	3	472

Usando el 50 % de los datos en el conjunto de entrenamiento y de prueba, obtenemos los siguientes valores por las metricas utilizadas:

Número	Score
0	0.8774
1	0.8318
2	0.9106
3	0.8704
4	0.8089
5	0.8861
6	0.8498
7	0.8359
8	0.8426
9	0.8446

Cuadro 1.4: Podemos observar que el número que tuvo un mejor puntaje en la predicción fue el 2, sin embargo la mayoría de los demás valores son menor a lo obtenido con el 70 % de los datos utilizados

En esta matriz de confusión podemos seguir observando que el número que mejor se clasifica es el 1 y podemos seguir observando que va disminuyendo el número de casos que se clasifica mal.

465	0	0	2	1	2	5	0	3	1
0	549	3	0	1	0	5	2	9	0
14	27	387	11	8	1	25	11	20	2
2	9	10	477	1	5	6	13	9	10
0	10	4	0	419	2	5	1	3	16
16	9	1	33	12	350	12	9	24	8
11	8	8	0	12	12	413	0	4	0
3	17	8	2	9	1	1	438	1	22
10	24	4	14	13	22	13	6	396	10
9	7	0	9	42	0	1	44	1	375

Podemos ver que el porcentaje para la selección del conjunto de entrenamiento y prueba, la mejor opción es del 90 %, pues obtenemos una mejor clasificación para cada número, por lo que trabajaremos con este

porcentaje de datos.

b)

Utilizando el 90 % de los datos del conjunto de entrenamiento y de prueba, obtenemos los siguiente:

Para LDA, obtenemos la siguiente precisión:

Clase	Precisión	Recall	F1-score	Soporte
0	0.94	0.96	0.95	888
1	0.89	0.96	0.92	1012
2	0.91	0.79	0.85	918
3	0.87	0.88	0.87	904
4	0.84	0.91	0.87	884
5	0.84	0.82	0.83	814
6	0.91	0.90	0.91	870
7	0.92	0.84	0.88	931
8	0.80	0.81	0.80	887
9	0.81	0.86	0.84	892

Cuadro 1.5: Podemos observar que los scores para la clasificación de LDA son muy buenos, lo que nos dice que es un buen modelo para clasificar.

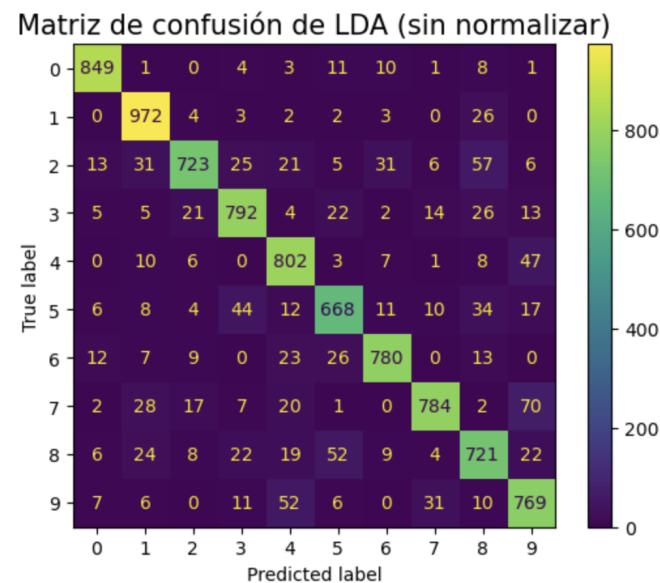


Figura 1.1: En esta matriz de confusión, podemos observar una baja incidencia de errores al clasificar, por lo que podemos decir que es un buen modelo para lo que se está realizando.

Para QDA, obtenemos la siguiente precisión:

Clase	Precisión	Recall	F1-score	Soporte
0	0.39	0.96	0.56	888
1	0.90	0.95	0.92	1012
2	0.89	0.24	0.38	918
3	0.71	0.42	0.53	904
4	0.95	0.21	0.35	884
5	0.87	0.17	0.28	814
6	0.62	0.96	0.76	870
7	0.93	0.36	0.52	931
8	0.47	0.61	0.53	887
9	0.48	0.94	0.63	892

Cuadro 1.6: Podemos observar que tiene una buena precision, a excepción de número 0, donde tuvo una precision muy baja a diferencia de los demás.

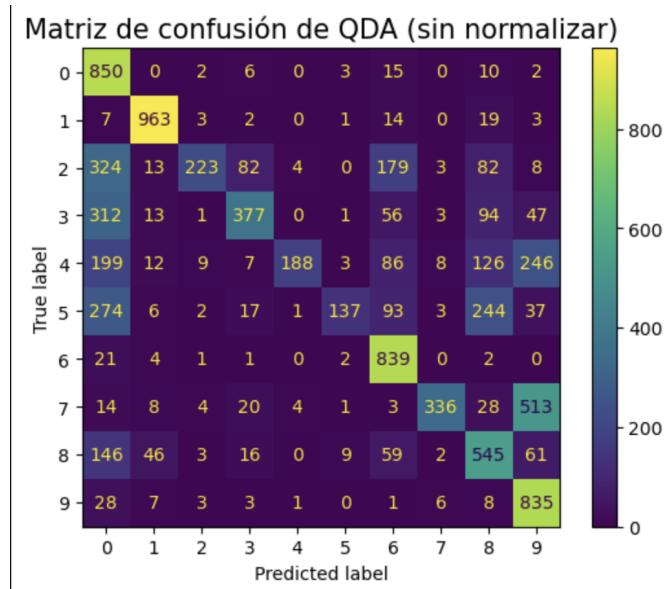


Figura 1.2: En esta matriz de confusión, podemos observar una alta incidencia de errores al clasificar, por lo que podemos decir que no es un buen modelo para lo que se está realizando.

c)

A continuación se muestran las capturas de las predicciones realizadas por los distintos modelos, en este caso se utilizaron los tres modelos vistos en los incisos anteriores.

Podemos observar que el modelo LDA tiene mayor número de aciertos, siendo el QDA el que peor clasifica, de igual forma podemos observar que el "baseline" que se pidió al inicio, clasifica tan bien como LDA, podríamos decir que para este tipo de problemas, un modelo lineal se ajusta bien al problema, en vez de un modelo cuadrático.



(a) Clasificación del número 7 dibujado



(b) Clasificación del número 5 dibujado



(c) Clasificación del número 3 dibujado



(d) Clasificación del número 4 dibujado



(e) Imagen 5



(f) Imagen 6

## 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 sí 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 estructuras latentes entre documentos y términos es usando análisis semántico latente (LSA), el cual se basa en factorizaciones apropiadas de esa matriz. Sea  $A_{m \times n}$  la matriz TF-IDF de rango  $r$ , con  $m$  renglones (documentos) y  $n$  columnas (términos). Una aproximación de rango  $k$  de esta matriz, está dada por la factorización SVD  $A \approx A(k) = U(k)\Sigma(k)V(k)^T$ , donde  $\Sigma(k)$  es diagonal con los  $k$  eigenvalores más grandes de  $A$  y  $U(k)$ ,  $V(k)$  contienen los correspondientes eigenvectores izquierdos y derechos que definen una base ortonormal para los espacios columna y renglón, respectivamente. Al aplicar esta 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  $V(k)$   $n \times k$  y  $U(k)$   $m \times k$  nos proporcionan una representación de los términos y documentos, respectivamente en términos de los  $k$  tópicos, y  $\Sigma(k)$  nos proporciona la importancia de cada tópico. En Python, puedes usar la implementación de `sklearn.decomposition.TruncatedSVD`. En este ejercicio, realizarás un análisis de tópicos en las transcripciones de las conferencias matutinas de la presidencia de México, las cuales puedes acceder en este repositorio. 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”.

- a) Obtén una representación TF-IDF de los textos. Define el tamaño del vocabulario y realiza el preprocesso 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.
- b) Obtén  $k$  tópicos mediante la descomposición SVD. Elige un  $k$  adecuado y justifícalo. Representa cada tópico mediante un word cloud de los términos que forman cada tópico según la importancia expresada en las magnitudes de los renglones de  $V(k)$ . ¿Puedes asignar un “nombre” representativo de cada tópico?
- 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  $XV(k)$  (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.
- 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  $U$  y  $V$ . Una forma de resolver este 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  $A$  de rango  $r$  con entradas no-negativas, NMF calcula una aproximación de rango  $k < r$  mediante la factorización  $A \approx A(k) = W(k)H(k)$ , donde  $W(k)$ ,  $H(k) \geq 0$ . En scikit-learn puedes usar la clase NMF del módulo `sklearn.decomposition.NMF`. Repite los incisos anteriores usando esta descomposición. ¿Cuál te parece mejor y por qué?

- 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 gráficarlos 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.

## 2.1. SOLUCIÓN (PROBLEMA 2)

- a) Haciendo uso del método 'procesamiento' dentro de la carpeta de ayudantias, con el cual quitamos todos los acentos que aparecen en el texto, tambien eliminamos los 'stop words' ya que no se consideran palabras clave para esta parte del procesamiento y podríamos considerarlo como ruido.

Para saber el tamaño de vocabulario a utilizar, obtuvimos el promedio de palabras por semana, el cual es de 25,981 palabras por semana, de las cuales 14,113 son 'stop words' por lo que el total de palabras significativas seria la diferencia de ambos promedios obtenidos. Dado que no es recomendable que el vocabulario sea tan grande, consideramos el tamaño de este de 500 palabras.

	abajo	acerca	actuar	acuerdo	adelante	ademas	adultos	\
0	0.011073	0.029756	0.007930	0.039195	0.015739	0.047035	0.005912	
1	0.016514	0.021516	0.013140	0.090928	0.039119	0.064949	0.014696	
2	0.004787	0.021049	0.022854	0.128773	0.022679	0.033888	0.015335	
3	0.000000	0.005719	0.011177	0.093916	0.027729	0.060769	0.000000	
4	0.010278	0.017575	0.024535	0.097013	0.029216	0.050932	0.005488	
..	..	..	..	..	..	..	..	..
254	0.003569	0.003487	0.006815	0.043789	0.020288	0.047157	0.026675	
255	0.006960	0.003400	0.026583	0.039417	0.046163	0.059125	0.007432	
256	0.012725	0.005329	0.001736	0.065202	0.036171	0.070349	0.003882	
257	0.011885	0.009678	0.011349	0.043006	0.043171	0.059834	0.000000	
258	0.022225	0.009872	0.000000	0.045772	0.047863	0.070565	0.025892	
	adversarios	aeropuerto	afortunadamente	...	ver	veracruz	\	
0	0.018936	0.065666	0.005287	...	0.135877	0.003073		
1	0.005379	0.014193	0.002628	...	0.148083	0.000000		
2	0.004678	0.017279	0.002285	...	0.119736	0.005314		
3	0.017157	0.060362	0.011177	...	0.082867	0.038987		
4	0.017575	0.058299	0.017174	...	0.143094	0.008558		
..	..	..	..	..	..	..	..	..
254	0.010461	0.000000	0.023852	...	0.165050	0.000000		
255	0.023803	0.057423	0.003323	...	0.154382	0.000000		
256	0.014210	0.022497	0.008679	...	0.154425	0.022200		
257	0.009678	0.053118	0.013241	...	0.158934	0.010996		
258	0.013820	0.008335	0.019293	...	0.167830	0.006730		

Figura 2.1: Representación TF-IDF de los textos obtenidos usando la parametrización mencionada

	verdad	vez	vida	viendo	viene	violencia	voy	\
0	0.010533	0.036582	0.047035	0.007869	0.028743	0.016044	0.107134	
1	0.020944	0.038969	0.033773	0.015648	0.010392	0.005317	0.111712	
2	0.009106	0.029369	0.038406	0.031750	0.004518	0.013872	0.133291	
3	0.016701	0.033147	0.044196	0.011092	0.005524	0.016960	0.066294	
4	0.021997	0.041231	0.033955	0.017043	0.007276	0.019856	0.060633	
..	..	..	..	..	..	..	..	..
254	0.033944	0.047157	0.030315	0.020288	0.020210	0.013788	0.141471	
255	0.049651	0.026278	0.042701	0.029676	0.003285	0.003361	0.045986	
256	0.015562	0.041180	0.048043	0.013780	0.017158	0.021071	0.049759	
257	0.015074	0.043006	0.020568	0.037540	0.018698	0.019135	0.065443	
258	0.019219	0.036236	0.047679	0.021060	0.026700	0.007807	0.051493	
	zona							
0	0.009256							
1	0.006135							
2	0.010670							
3	0.006523							
4	0.005728							
..	..							
254	0.007955							
255	0.000000							
256	0.006078							
257	0.011039							
258	0.013512							

Figura 2.2: Representación TF-IDF de los textos obtenidos usando la parametrización mencionada

- b) Haciendo uso de una grafica de varianza explicada acumulada, obtenemos el valor adecuado de  $k$  para poder aplicar la descomposicion SVD, como se muestra en la siguiente imagen:

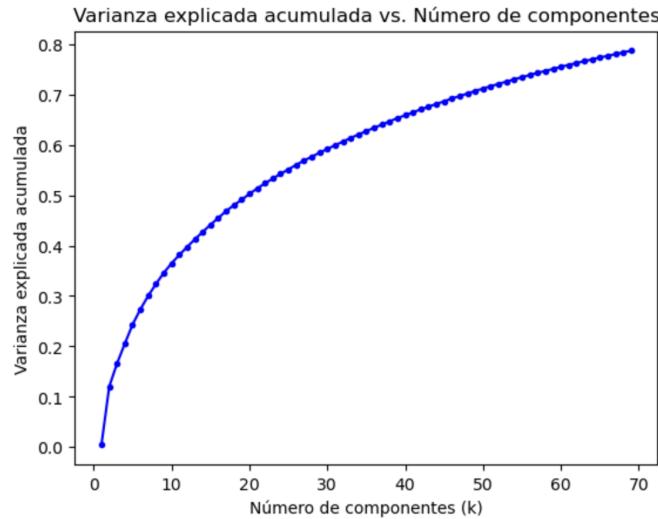
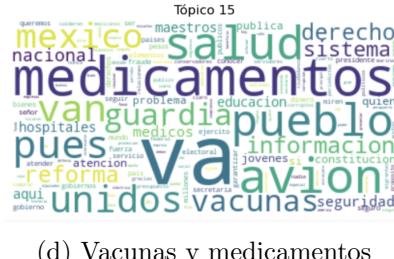
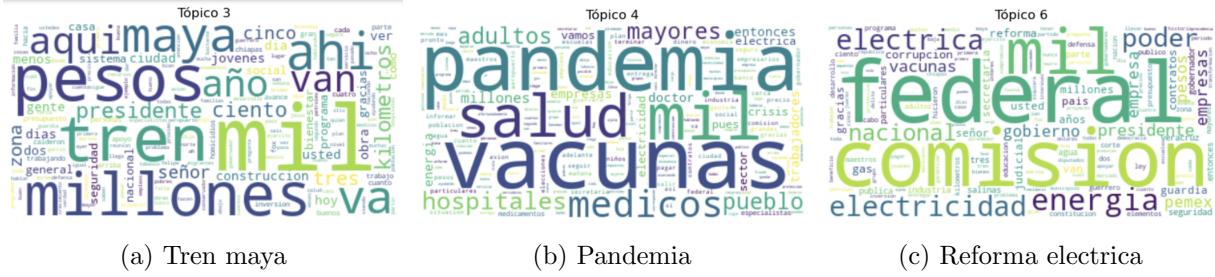
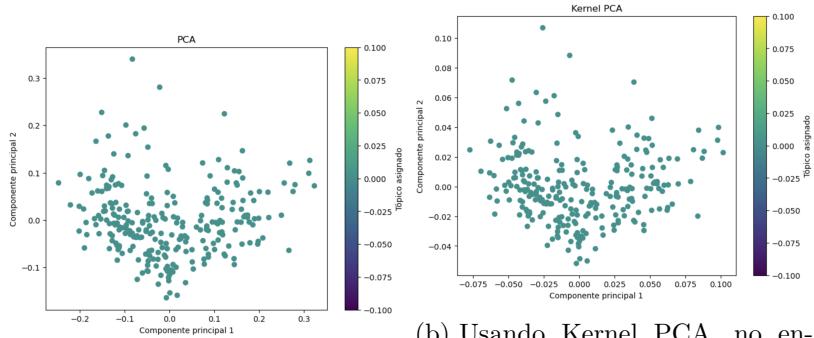


Figura 2.3: Podemos observar que a partir de  $k = 50$  se obtiene más del setenta porciento de la varianza explicada acumulada, sin embargo a la hora de generar las 'word clouds', no fue posible asignarles un 'nombre', por lo que nos quedamos con  $k = 20$  pues de esta forma si nos fue posible asignar a la mayoria de los topics un 'nombre'

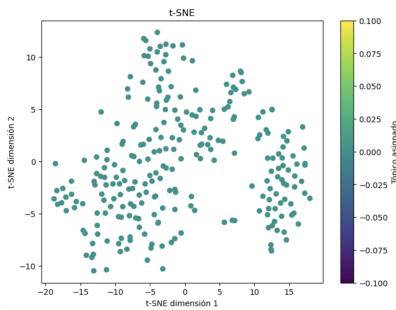
Algunos de los tópicos a los cuales se les pudo asignar un 'nombre' son:



- c) Usando una visualización de baja dimensión (2 componentes), obtenemos lo siguiente:

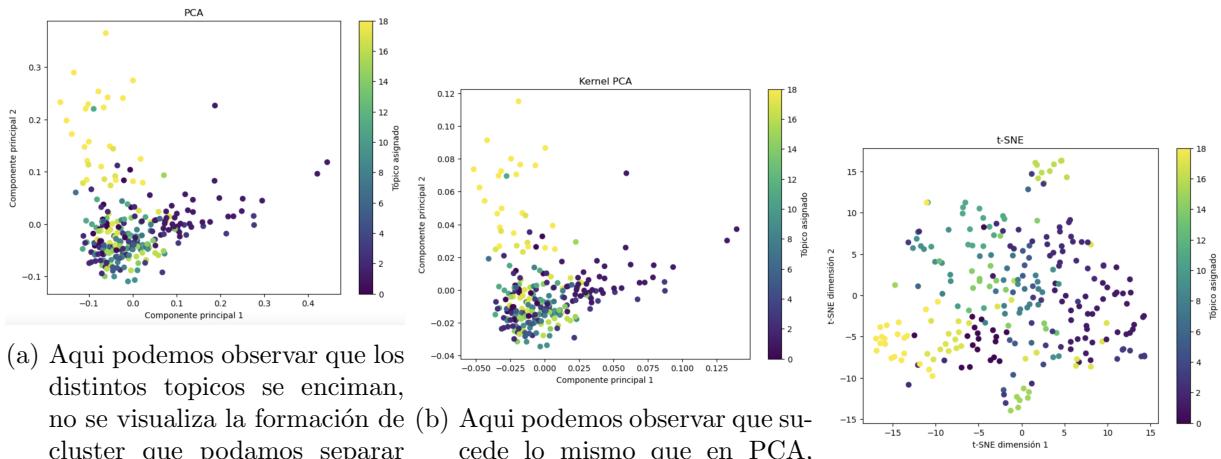


- (a) Usando PCA, no encontramos patrones interesantes, lo toma como si fueran un único tópico
- (b) Usando Kernel PCA, no encontramos patrones interesantes, lo toma como si fueran un único tópico



- (c) Usando t-SNE, no encontramos patrones interesantes, lo toma como si fueran un único tópico

- d) Haciendo uso de la factorización no negativa de matrices (NMF), obtenemos lo siguiente:



(a) Aquí podemos observar que los distintos topicos se enciman, no se visualiza la formación de cluster que podamos separar con facilidad, solo unos cuantos de topicos de una misma clase se separan y forman pequeños grupos aunque hay presencia de otros tipos junto a ellos.

(b) Aquí podemos observar que sucede lo mismo que en PCA, los distintos tipos de topicos se juntan y solo unos cuantos de topicos de una misma clase se separan y forman pequeños grupos aunque hay presencia de otros tipos junto a ellos.

(c) Aquí sin embargo podemos ver que los distintos tipos de topicos ya no se juntan tanto, por lo que es más facil observar de que tipo de topico son y podríamos formar pequeños clusters.