Centro de Investigación en Matemáticas (CIMAT).
Unidad Monterrey

Ciencia de Datos Tarea 3

Diego Godinez Bravo

22 de abril de 2024

1. Problema 1

Considera los datos MNIST de dígitos escritos a mano que usamos anteriormente de 28×28 pixeles. Para mayor facilidad, puse los datos en archivos csv (mnist.zip): (`mnist_Xtrain.csv`, `mnist_Ytrain.csv`) contienen los valores de los pixeles (normalizados) y su respectiva categoría para entrenamiento, y (`mnist_Xtest.csv`, `mnist_Ytest.csv`), lo mismo para los datos de prueba. La Figura 1 muestra un ejemplo de estos datos, el cual se generó con el Código MNIST.

En este ejercicio implementarás métodos de clasificación para los $k \in K = \{0, 1, ..., 9\}$ 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 $(x_{\text{train}}, y_{\text{train}})$, $(x_{\text{test}}, y_{\text{test}})$ 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.

1.1. Solución

1.1.1. Métricas de Desempeño Clasificador Basado en el Baseline Implementado.

Las métricas de desempeño porporcionan un panorama general del rendimiento del modelo en cada clase, además de que permiten comparar diferentes modelos y seleccionar el más adecuado para una tarea específica.

Se observa una alta precisión (*precision*) en la mayoría de las clases, esta métrica nos ayuda a cuantificar la fracción de positivos verdaderos entre el total de los clasificados como positivos, es decir, nos indica que cuando se predice una etiqueta positiva en la mayoría de los casos es correcta.

La recuperación (*recall*) varía significativamente entre las clases. Para este modelo, las clases 0, 1 y 6 muestran un valor alto, lo que significa que el modelo identifica correctamente la mayoría de las instancias positivas en esas clases.

Por último, el *F1-score* resume la evaluación del modelo relacionando la precisión y la recuperación. En este caso, los valores obtenidos indican un buen equilibrio entre estas dos métricas en cada una de las clases.

	Precision	Recall	F1-score	Support
0	0.91	0.96	0.93	980
1	0.84	0.98	0.90	1135
2	0.92	0.79	0.85	1032
3	0.85	0.87	0.86	1010
4	0.81	0.90	0.85	982
5	0.88	0.74	0.80	892
6	0.88	0.91	0.89	958
7	0.85	0.86	0.86	1028
8	0.84	0.78	0.81	974
9	0.84	0.79	0.82	1009
accuracy			0.86	10000
macro avg	0.86	0.86	0.86	10000
weighted avg	0.86	0.86	0.86	10000

El modelo muestra una exactitud (*accuracy*) del 86 %, lo cual indica que el 86 % de los datos son clasificados correctamente. Sin embargo es importante señalar que si los datos están desbalanceados o si se está más interesado en detectar una de las clases, la exactitud no captura realmente la eficacia del modelo.

Al evaluar globalmente las métricas obtenidas, junto con la exactitud del modelo, podemos concluir que éste muestra un buen rendimiento al momento de clasificar las distintas clases.

La matriz de confusión muestra la distribución de las predicciones del modelo en comparación con las clases reales. Los elementos en la diagonal representan el número de puntos para los cuales la etiqueta predicha es igual a la etiqueta verdadera, por otro lado, los elementos fuera de la diagonal son aquellos que el clasificador etiqueta de manera incorrecta.

A través de la martiz de confusión observamos que la clase 5 registra el mayor número de predicciones incorrectas, principalmente confundidas con la clase 3. Además, la clase 8 también presenta una proporción notable de predicciones erróneas, mayormente confundidas con las clases 1 y 5. Sin embargo, en la mayoría de las clases, el número de predicciones incorrectas es relativamente bajo, lo que respalda el rendimiento y la precisión del modelo en la clasificación (**Figura 1.1)**.

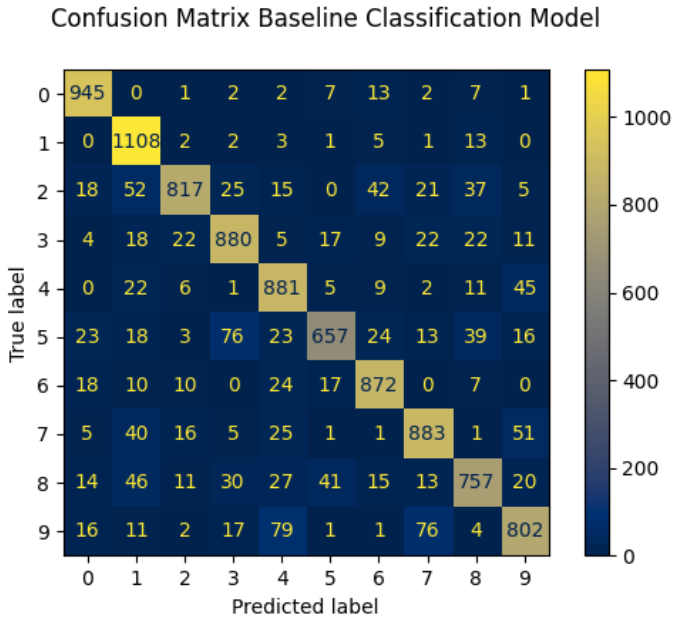

Figura 1.1. Matriz de confusión del modelo de clasificación basado en el baseline implementado.

1.2. Solución

1.2.1. Métricas de Desempeño Clasificadores Basados en LDA y QDA.

El análisis discriminante lineal (LDA) es un método utilizado en el aprendizaje automático supervisado para resolver problemas de clasificación de múltiples clases. De manera general, LDA separa múltiples clases con múltiples características mediante la reducción de la dimensionalidad de los datos.

En comparación con los resultados obtenidos a partir del baseline implementado, el modelo de clasificación basado en LDA resulta en un mejor rendimiento al momento de predecir las etiquetas en este conjunto de datos en particular. Esto basándonos en la precisión, recuperación, y en el *F1-score*.

	Precision	Recall	F1-score	Support
0	0.94	0.96	0.95	980
1	0.89	0.97	0.93	1135
2	0.92	0.79	0.85	1032
3	0.87	0.87	0.87	1010
4	0.84	0.90	0.87	982
5	0.84	0.82	0.83	892
6	0.91	0.89	0.90	958
7	0.91	0.84	0.88	1028
8	0.80	0.81	0.80	974
9	0.81	0.85	0.83	1009
accuracy			0.87	10000
macro avg	0.87	0.87	0.87	10000
weighted avg	0.87	0.87	0.87	10000

El modelo muestra una exactitud del 87 %, es decir, el 87 % de los datos son clasificados correctamente. A diferencia del 86 % obtenido a partir del modelo de clasificación basado en el baseline implementado.

En este caso el número de predicciones incorrectas en las clases 5 y 8 disminuyeron en gran medida, esto en comparación con el modelo de clasificación anterior. Sin embargo, estos siguen siendo las clases que poseen el mayor número de predicciones incorrectas. Observamos que en las clases 8 y 9 tambien disminuyeron en gran medida el número de predicciones incorrectas.

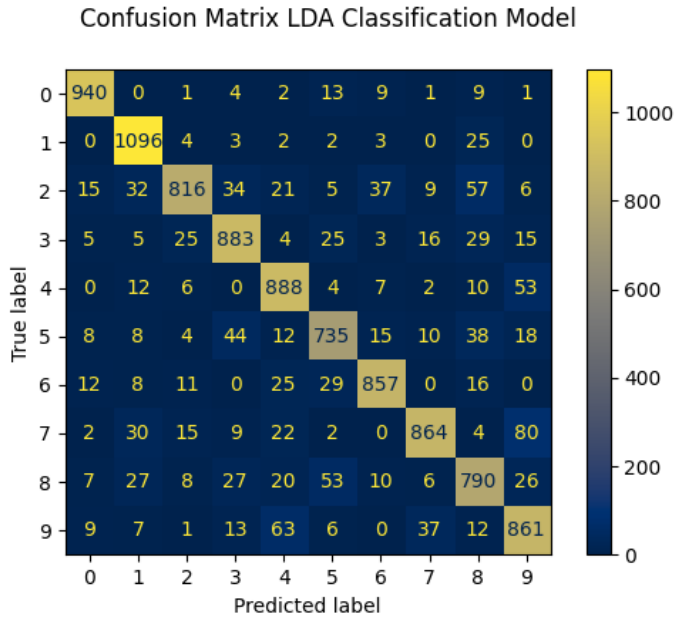

Figura 1.2. Matriz de confusión del modelo de clasificación basado en LDA.

El análisis discriminante cuadrático (QDA) es un método estadístico utilizado en el aprendizaje automático supervisado para la clasificación de datos. En contraste con el análisis discriminante lineal (LDA), QDA no asume que las covarianzas de las clases son iguales, lo que resulta en la captura de relaciones más complejas entre las características de los datos y las clases a las que pertenecen.

A diferencia de los resultados obtenidos al evaluar el rendimiento de los modelos basados en el baseline implementado y en LDA, el modelo de clasificación basado en QDA posee un bajo rendimiento al predecir las etiquetas para este conjunto de datos en específico. Esto lo podemos afirmar debido a que los valores obtenidos para cada una de las métricas son extremadamente bajos en comparación con los modelos previamente ajustados.

	Precision	Recall	F1-score	Support
0	0.33	0.96	0.49	980
1	0.88	0.96	0.91	1135
2	0.86	0.24	0.37	1032
3	0.69	0.36	0.47	1010
4	0.93	0.22	0.35	982
5	0.86	0.28	0.42	892
6	0.71	0.95	0.82	958
7	0.91	0.37	0.52	1028
8	0.52	0.48	0.50	974
9	0.48	0.94	0.64	1009
accuracy			0.58	10000
macro avg	0.72	0.57	0.55	10000
weighted avg	0.72	0.58	0.55	10000

La afirmación anterior se respalda con el valor de la exactitud obtenido, el cual es del 58 %, lo cual se puede observar con mayor claridad en la matriz de confusión obtenida **Figura 1.3**. El número de predicciones incorrectas aumentan en gran medida para las clases la mayoría de las clases (clase 2, clase 3, clase 4, clase 5, clase 7 y clase 8), lo que coincide con la exactitud del modelo, donde solo el 58 % de los datos son clasificados correctamente.

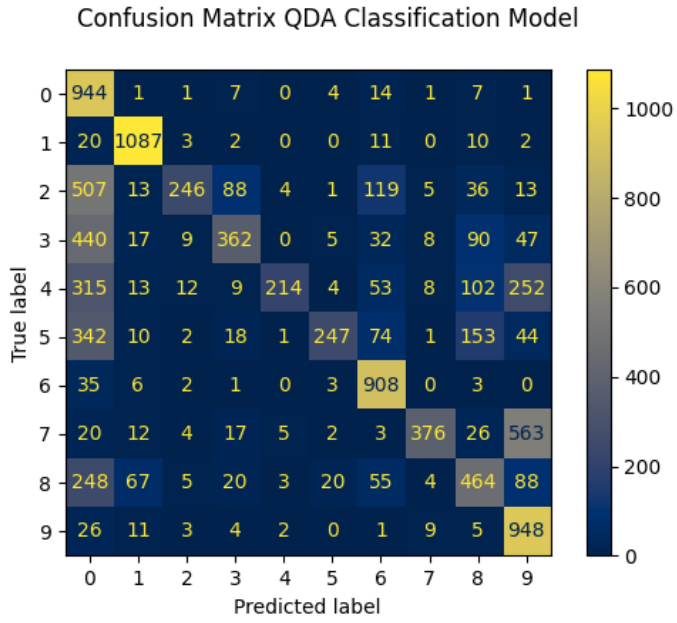

Figura 1.3. Matriz de confusión del modelo de clasificación basado en QDA.

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 $A \approx A(k) = U(k)\Sigma(k)V(k)'$, 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, respectívamente. 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 $V(k)$ es una matriz $n \times k$ y $U(k)$ es una matriz $m \times k$ que 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. 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, contenidas 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 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.

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

Term-Frequency-Inverse Document Frequency (TF-IDF) es una medida estadística que muestra la relevancia de palabras clave en un conjunto de documentos.

TF-IDF es una combinación de dos conceptos diferentes: *term frequency*, e *inverse document frequency*. *Term Frequency* se utiliza para medir el número de veces que un término está presente en un documento. Por otro lado, *Inverse Document Frequency* asigna pesos a las palabras según su frecuencia en un documento, asignando un menor peso a las palabras frecuentes y un mayor peso a las palabras infrecuentes. En general, la medida TF-IDF no es más que la multiplicación de ambos valores.

En este caso se analizaron las transcripciones

Con el objetivo de mejorar la calidad, entendimiento y relevancia de las transcripciones se implementaron una serie de transformaciones de preprocesamiento de texto.

- En primera instancia se realizó la conversión de todos los caracteres del texto en minúsculas. Esto con el fin de mejorar la coherencia y brindar consistencia en el análisis.

- Posteriormente se eliminaron los caracteres especiales, signos de puntuación, acentos, espacios y dígitos. Esto ayuda a mejorar la claridad del texto, brindar consistencia y mantener solo los caracteres relevantes. En el caso de los dígitos, estos pueden resultar irrelevantes al momento de realizar el análisis del texto, por lo que se eliminan para enfocar los esfuerzos en obtener información relevante de las palabras.

- Se eliminaron las palabras formadas por 3 caracteres o menos. Por lo general, estas palabras no aportan relevancia al texto y pueden contribuir ruido en el análisis.

- Por último, se eliminaron las palabras vacías (*stop words*). Las palabras vacías son términos que ocurren con alta frecuencia pero que no aportan mucho significado en el análisis del texto.

2.2. Solución

A partir de la descomposición SVD, se obtuvieron k tópicos los cuales se representaron mediante un word cloud. Los tópicos generados con $k < 4$ están conformados por palabras que describen en gran medida cada uno de los k tópicos.

En el tópico 1 el tema representativo está vinculado a aspectos de seguridad y defensa nacional, en este caso, un nombre apropiado podría ser 'Seguridad Nacional'. Por otro lado, el word cloud ligado al tópico 2 esta relacionado con temas del sector salud, sugiriendo el nombre del tópico como 'Salud Pública'. Posteriormente el word cloud generado para el tópico 3 incluye palabras relacionadas a temas de energía y combustibles fósiles, por lo que un nombre adecuado podría ser 'Energías y Combustibles'. Por último, el tópico 4 se encuentra relacionado con los principales proyectos de desarrollo en el país, sugiriendo el tema adecuado como 'Infraestructura'.

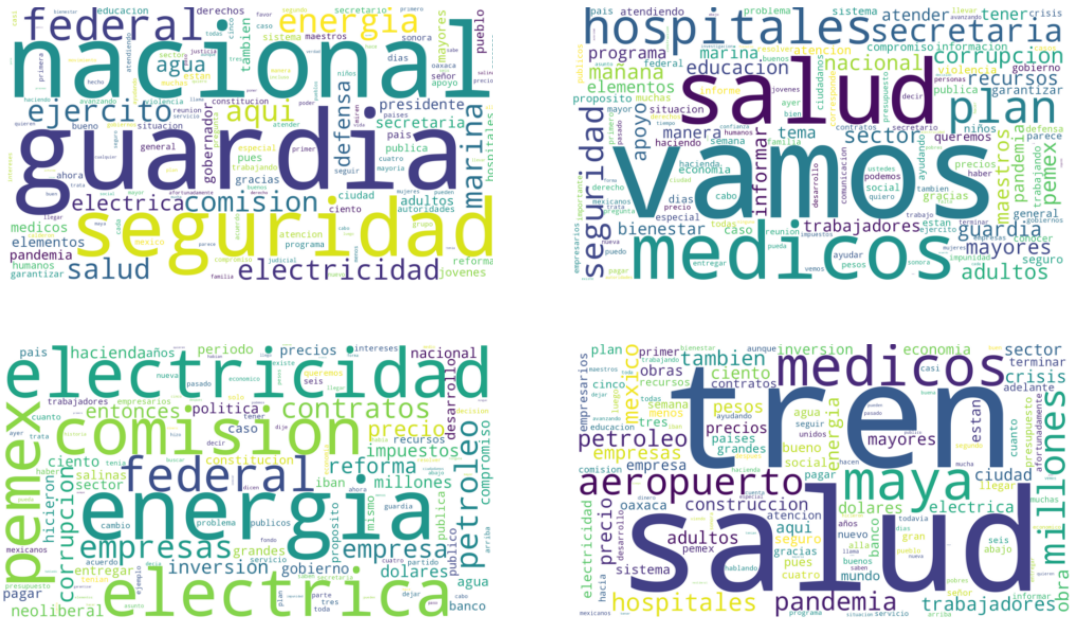

Figura 2.1. Conjunto de cloud words generados con $k \leq 4$.

Es importante mencionar que en cada uno de los word clouds generados se incluyen palabras fuera del tópico principal, lo que podría dificultar la identificación de los tópicos y la asignación de un 'nombre' principal a cada uno de ellos.

A partir de $k > 4$ se observa cierto grado de redundancia en los word clouds generados. Además, a diferencia de los casos anteriores, no se puede asignar un nombre representativo a los tópicos, ya que las palabras que los conforman no son muy descriptivas ni informativas sobre el tema principal de cada uno de ellos.

Figura 2.2. Conjunto de cloud words generados a partir de $k > 4$.

2.3. Solución

Las transcipciones se asignaron en su mayoría a un solo tópico, esto puede ser consecuencia de que los temas de debate en las conferencias presidenciales se hayan reiterado con una alta frecuencia. Sin embargo, en este caso es más probable que esto sea resultado de una limitación en la capacidad del modelo para capturar la diversidad de tópicos existentes en las transcripciones. Considerando lo anterior, si el modelo no logra identificar de manera precisa los distintos tópicos, los métodos de reducción de dimensionalidad y visualización podrían presentar dificultades para identificar y representar patrones significativos (**Figura 2.3**).

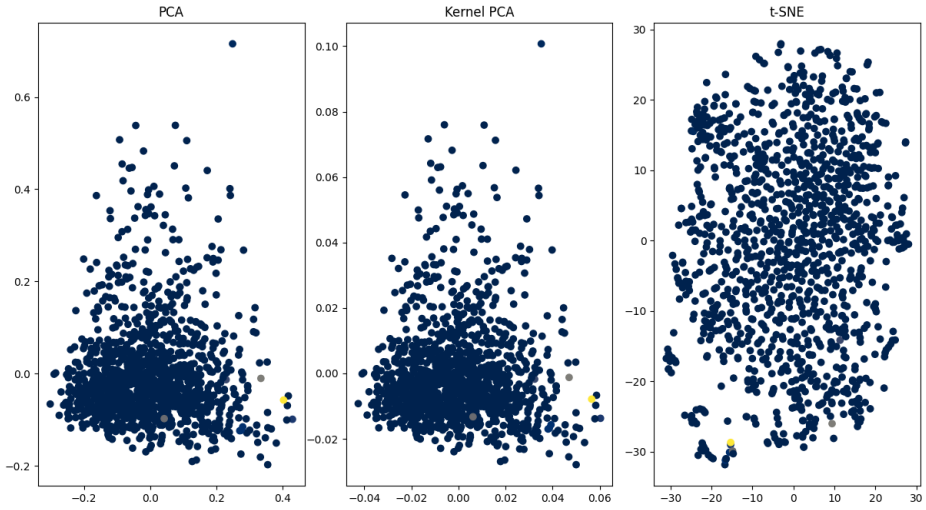

Figura 2.3. Visualizaciones de baja dimensión basadas en PCA, Kernel-PCA y t-SNE.

2.4. Solución

De manera general observamos una mejora significativa en las cloud words generadas, donde la identificación de los tópicos es más clara y la tarea de asignr un 'nombre' descriptivo resulta más sencilla.

Esta mejora puede atribuirse al empleo de la factorización de matriz no negativa (NMF) para la obtención de los tópicos. Nos permite obtener tópicos más coherentes y descriptivos, lo que resulta en una representación visual más clara en las word clouds generadas (**Figura 2.4**).

Figura 2.4. Conjunto de cloud words generados a partir de k tópicos obtenidos mediante la factorización NMF.

En general, observamos una mayor coherencia y distinción en los temas representados en las word clouds generadas a partir de los tópicos obtenidos mediante NMF. Las palabras clave relacionadas con cada tópico tienden a estar más concentradas y son más representativas y descriptivas de los temas discutidos en las conferencias presidenciales. Esto facilita una interpretación más precisa y una comprensión más profunda de los temas abordados en el discurso presidencial.

Las transcipciones se asignaron a diferentes tópicos gracias al empleo del método NMF para la obtención de los k tópicos, se mejoró de manera significativa la identificación de los tópicos y de las palabras asociadas a cada uno de ellos (**Figura 2.5**).

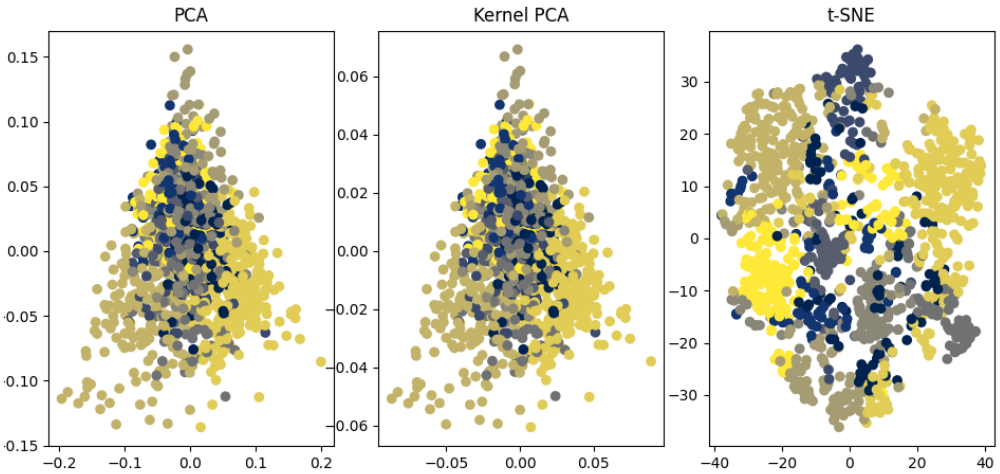

Figura 2.5. Visualizaciones de baja dimensión basadas en PCA, Kernel-PCA y t-SNE.