Taller 2

1. **Análisis del Artículo de Zou, H., Hastie, T., & Tibshirani, R. (2006)**

   a. **Explicación de la Propuesta de los Autores (SPCA)**
   El artículo "Sparse Principal Component Analysis" de Zou, Hastie y Tibshirani (2006) introduce una modificación al Análisis de Componentes Principales (PCA) tradicional con el objetivo de mejorar la interpretabilidad de los componentes. El PCA estándar sufre de la limitación de que cada componente principal (CP) es una combinación lineal de *todas* las variables originales, lo que dificulta asignar un significado concreto a dichos componentes, especialmente cuando el número de variables es grande.

   La propuesta central de los autores, denominada Análisis de Componentes Principales Disperso (SPCA), consiste en obtener componentes principales cuyos vectores de *loadings* (cargas o coeficientes) sean *dispersos*, es decir, que contengan muchos ceros. Esto implica que cada componente disperso se define por un subconjunto reducido de las variables originales, facilitando así su interpretación.

   Para lograr esta dispersión, el paper reformula ingeniosamente el problema de PCA como un problema de regresión. Los teoremas clave (Teorema 1, 2 y 3 del artículo) establecen que los loadings de PCA pueden obtenerse minimizando un criterio de reconstrucción de los datos sujeto a una penalización de tipo Ridge (L_2) sobre los loadings. Una vez establecida esta equivalencia, los autores proponen modificar este criterio de optimización reemplazando o complementando la penalización Ridge con una penalización de tipo Lasso (L_1) o Elastic Net (una combinación de L_1 y L_2). La penalización L_1 es fundamental, ya que tiene la propiedad de encoger algunos coeficientes exactamente a cero, logrando así la selección de variables y la dispersión deseada en los loadings.

   Fundamentos Teóricos Clave

   - **Aproximaciones Directas Dispersas (Sección 3.1 del paper)**
   Inicialmente, se muestra una aproximación en dos etapas. El Teorema 1 establece la conexión entre PCA y regresión Ridge:

   Teorema 1 (PCA como Regresión Ridge - Teorema 1 del paper)**.** *Sea* $\mathbf{Z}_i = \mathbf{U}_i D_{ii}$ *el i-ésimo componente principal. Para* λ¿0*, los estimadores Ridge* $\hat{\boldsymbol{\beta}}_{ridge}$ *dados por:*

$$\hat{\boldsymbol{\beta}}_{ridge} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{Z}_i - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2 \tag{1}$$

   cumplen que $\hat{\boldsymbol{\beta}}_{ridge}/\|\hat{\boldsymbol{\beta}}_{ridge}\|_2 = \mathbf{V}_i$ *(el i-ésimo vector de loadings de PCA).*

A partir de esto, se pueden obtener loadings dispersos añadiendo una penalización L_1 (γ) al problema de regresión:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{Z}_i - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2 + \gamma\|\boldsymbol{\beta}\|_1 \tag{2}$$

y luego normalizando $\tilde{\mathbf{V}}_i = \hat{\boldsymbol{\beta}}/\|\hat{\boldsymbol{\beta}}\|_2$.

- **Criterio SPCA Unificado (Sección 3.2 del paper)**
  El enfoque principal es un criterio unificado que busca simultáneamente la matriz de scores $\mathbf{A} \in \mathbb{R}^{n\times k}$ y la matriz de loadings dispersos $\mathbf{B} \in \mathbb{R}^{p\times k}$. Los Teoremas 2 y 3 son cruciales aquí:

  Teorema 2 (Primer CP como Regresión - Teorema 2 del paper)**.** *Para λ¿0, la solución $(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}})$ al problema:*

$$(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}) = \arg\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \sum_{i=1}^{n} \|\mathbf{x}_i^T - \alpha_i\boldsymbol{\beta}^T\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2 \quad \textit{sujeto a } \|\boldsymbol{\alpha}\|_2^2 = 1 \tag{3}$$

  implica que $\hat{\boldsymbol{\beta}}$ es proporcional a $\mathbf{V}_1$.
  Definiciones (Teorema 2): $\mathbf{x}_i^T$ es la i-ésima observación, $\boldsymbol{\alpha}$ el vector de scores, $\boldsymbol{\beta}$ el vector de loadings. La ecuación minimiza el error de reconstrucción con una penalización Ridge sobre $\boldsymbol{\beta}$, y normaliza los scores.

  Teorema 3 (Primeros k CPs como Regresión - Teorema 3 del paper)**.** *Para λ¿0, la solución $(\hat{\mathbf{A}}, \hat{\mathbf{B}})$ al problema:*

$$(\hat{\mathbf{A}}, \hat{\mathbf{B}}) = \arg\min_{\mathbf{A},\mathbf{B}} \|\mathbf{X} - \mathbf{A}\mathbf{B}^T\|_F^2 + \lambda\sum_{j=1}^{k} \|\boldsymbol{\beta}_j\|_2^2 \quad \textit{sujeto a } \mathbf{A}^T\mathbf{A} = \mathbf{I}_k \tag{4}$$

  implica que las columnas $\hat{\boldsymbol{\beta}}_j$ de $\hat{\mathbf{B}}$ son proporcionales a los loadings $\mathbf{V}_j$ del PCA tradicional.
  Definiciones (Teorema 3): $\mathbf{X}$ es la matriz de datos, $\mathbf{A}$ la matriz de scores ($n \times k$), $\mathbf{B}$ la matriz de loadings ($p \times k$). Se minimiza el error de reconstrucción global con penalización Ridge sobre cada vector de loading $\boldsymbol{\beta}_j$, y se impone ortonormalidad a los scores ($\mathbf{A}^T\mathbf{A} = \mathbf{I}_k$).
  El criterio SPCA final para k componentes dispersos se define entonces como:

$$(\hat{\mathbf{A}}, \hat{\mathbf{B}}) = \arg\min_{\mathbf{A},\mathbf{B}} \|\mathbf{X}-\mathbf{A}\mathbf{B}^T\|_F^2+\lambda_2\sum_{j=1}^{k} \|\boldsymbol{\beta}_j\|_2^2+\sum_{j=1}^{k} \lambda_{1,j}\|\boldsymbol{\beta}_j\|_1 \quad \text{sujeto a } \mathbf{A}^T\mathbf{A} = \mathbf{I}_k \tag{5}$$

  donde λ_2 es la penalización L_2 (Ridge) y $\lambda_{1,j}$ es la penalización L_1 (Lasso) para el j-ésimo componente, permitiendo un control individual de la dispersión.

- **Solución Numérica y Varianza Ajustada (Secciones 3.3 y 3.4)**
  El criterio SPCA (5) se resuelve mediante un algoritmo alternante:
  i. Se inicializa $\mathbf{A}$.
  ii. Se itera hasta convergencia:
    A. Fijando $\mathbf{A}$, se resuelve para $\mathbf{B}$ (cada $\boldsymbol{\beta}_j$ se obtiene mediante un problema de regresión Elastic Net).

B. Fijando $\mathbf{B}$, se resuelve para $\mathbf{A}$ (minimizando $\|\mathbf{X} - \mathbf{A}\mathbf{B}^T\|_F^2$ sujeto a $\mathbf{A}^T\mathbf{A} = \mathbf{I}_k$, lo cual se logra mediante una rotación de Procrustes).

iii. Los loadings finales se normalizan.

Dado que los componentes dispersos pueden estar correlacionados, el paper también propone una fórmula para calcular la *varianza total explicada ajustada*, que tiene en cuenta estas correlaciones, usualmente mediante una descomposición QR de la matriz de scores $\tilde{\mathbf{Z}} = \mathbf{X}\tilde{\mathbf{B}}$.

En resumen, SPCA ofrece un marco metodológico robusto para obtener componentes principales interpretables mediante la inducción de dispersión en los loadings, basándose en la reformulación de PCA como un problema de regresión penalizada.

b. **Replicación del Ejemplo PITPROPS DATA**
Para replicar el ejemplo "PITPROPS DATA" del artículo, se utilizó Python junto con librerías como pandas para la manipulación de datos, pyreadr para cargar el dataset en formato .rda (R Data File), scikit-learn para las implementaciones de PCA y SparsePCA, y matplotlib/seaborn para la visualización.

Metodología de Implementación Resumida La implementación buscó aproximar los resultados y las visualizaciones presentadas en el paper, especialmente la Figura 2, que muestra la relación entre el Porcentaje de Varianza Explicada (PEV) y el parámetro de regularización λ_1 (proxy alpha en scikit-learn) para cada uno de los primeros seis componentes principales.

i. **Carga y Preprocesamiento de Datos:** El dataset Pitprops fue cargado desde un archivo .rda. Posteriormente, los datos fueron estandarizados (centrados y escalados a varianza unitaria) para asegurar que todas las variables contribuyeran de forma equitativa al análisis. La varianza total de los datos estandarizados (VarTotal(X)) se calculó como referencia para el PEV.

ii. **PCA Tradicional de Referencia:** Se realizó un PCA estándar sobre los datos estandarizados para obtener los valores de PEV de los primeros seis componentes principales. Estos valores sirven como línea base de comparación.

iii. **Generación de Curvas PEV vs. Alpha (Aproximación Figura 2):** Para cada uno de los seis componentes potenciales (de $j = 1$ a 6):
- Se iteró sobre un rango predefinido de valores para el parámetro alpha (actuando como $\lambda_{1,j}$).
- En cada iteración, se ajustó un modelo SparsePCA con $n_components$=1 sobre los datos originales estandarizados. (Nota: El paper no es explícito sobre si se realiza deflación secuencial para generar estas curvas; aquí se asume una aproximación a cada PC original independientemente).
- Se calcularon los scores ($Z_j = X\beta_j$) para este único componente disperso y su varianza (Var(Z_j)).
- El PEV se obtuvo como PEV$_j$ = Var(Z_j)/VarTotal(X).
- Se registró el PEV y el número de loadings no-cero para cada alpha.

Estos datos se utilizaron para graficar PEV vs. alpha para cada componente, similar a la Figura 2 del paper.

iv. **Selección de α_j y Construcción de SPCA Final:** A partir de las curvas generadas, se seleccionó un valor de α_j para cada componente j. La estrategia de selección buscó principalmente igualar el número de loadings no-cero reportados en la Tabla 3 del paper para cada componente. Si múltiples alpha producían el número deseado de no-ceros, se escogió aquel que maximizaba el PEV. Si no se encontraba el número exacto, se tomaba el más cercano. Con los α_j elegidos, se obtuvieron los loadings finales β_j para cada componente (utilizando los loadings ya calculados en el paso anterior para el alpha seleccionado). Estos loadings se ensamblaron en la matriz final $\mathbf{B}_{\text{final\_spca}}$.

v. **Cálculo de Scores y Varianza Ajustada Final:** Los scores finales se calcularon como $\mathbf{Z}_{\text{final\_spca}} = \mathbf{X}_{\text{scaled}}\mathbf{B}_{\text{final\_spca}}$. Dado que estos componentes, construidos "individualmente", no están garantizados de ser ortogonales, la varianza explicada ajustada se calculó utilizando la descomposición QR de $\mathbf{Z}_{\text{final\_spca}}$ (centrada), como sugiere el paper. La j-ésima varianza ajustada es R_{jj}^2/n, donde R_{jj} es el j-ésimo elemento diagonal de la matriz $\mathbf{R}$ y n es el número de observaciones.

vi. **Visualización y Comparación:** Se generaron heatmaps para los loadings del PCA tradicional y del SPCA final, y tablas comparativas resumiendo el número de no-ceros y la varianza explicada ajustada en comparación con los valores reportados en el paper.

Resultados Obtenidos (Ejemplo de Salida) A continuación, se presenta un extracto de los resultados obtenidos durante la ejecución, que incluye el PEV del PCA tradicional, la selección de alpha para cada componente SPCA, y una tabla comparativa final.

- **PEV de PCA Tradicional (Referencia)**

```
PC1: 46.01%
PC2: 20.65%
PC3: 16.16%
PC4: 7.54%
PC5: 4.34%
PC6: 2.55%
```

- **Selección de Alpha y Construcción de SPCA Final (Extracto)**

```
Construyendo SPCA Final con Alphas Elegidos Individualmente
* Componente 1: Alpha elegido = 1.712 -$>$ PEV = 39.49%, No-ceros = 7
* Componente 2: Alpha elegido = 2.629 -$>$ PEV = 29.95%, No-ceros = 5 (Pa
* Componente 3: Alpha elegido = 2.629 -$>$ PEV = 29.95%, No-ceros = 5 (Pa
* Componente 4: Alpha elegido = 3.257 -$>$ PEV = 7.69%, No-ceros = 1
* Componente 5: Alpha elegido = 3.257 -$>$ PEV = 7.69%, No-ceros = 1
* Componente 6: Alpha elegido = 3.257 -$>$ PEV = 7.69%, No-ceros = 1
```

Nota: Para los componentes 2 y 3, el número de no-ceros objetivo (4)

no se alcanzó exactamente con la discretización de alpha utilizada, seleccionándose el más cercano (5 no-ceros).

- **Resumen Comparativo Final con Alphas Individuales** La Tabla 1 muestra que, si bien el número de loadings no-cero se aproxima a los valores del paper, la varianza ajustada explicada por la implementación puede diferir. Esto se debe a múltiples factores, incluyendo las diferencias entre la implementación SparsePCA de scikit-learn y el algoritmo exacto del paper, la selección de alpha y la no ortogonalidad garantizada de los componentes construidos secuencialmente de esta manera. La varianza ajustada acumulada de la implementación (41.62%) es inferior a la del paper (75.7%).

Table 1. Comparación de SPCA: Implementación vs. Paper (PITPROPS)

Componente	Alpha Elegido	No-Ceros (Impl.)	No-Ceros (Paper)	PEV Ind. (Impl. %)	Var. Adj. (Impl. %)	Var. Adj. (Paper %)	Var. Adj. Acum. (Impl. %)	Var. Adj. Acum. (Paper %)
SPC1	1.71	7	7	39.49	39.49	28.0	39.49	28.0
SPC2	2.63	5	4	29.95	1.58	14.0	41.07	42.0
SPC3	2.63	5	4	29.95	0.00	13.3	41.07	55.3
SPC4	3.26	1	1	7.69	0.55	7.4	41.62	62.7
SPC5	3.26	1	1	7.69	0.00	6.8	41.62	69.5
SPC6	3.26	1	1	7.69	0.00	6.2	41.62	75.7

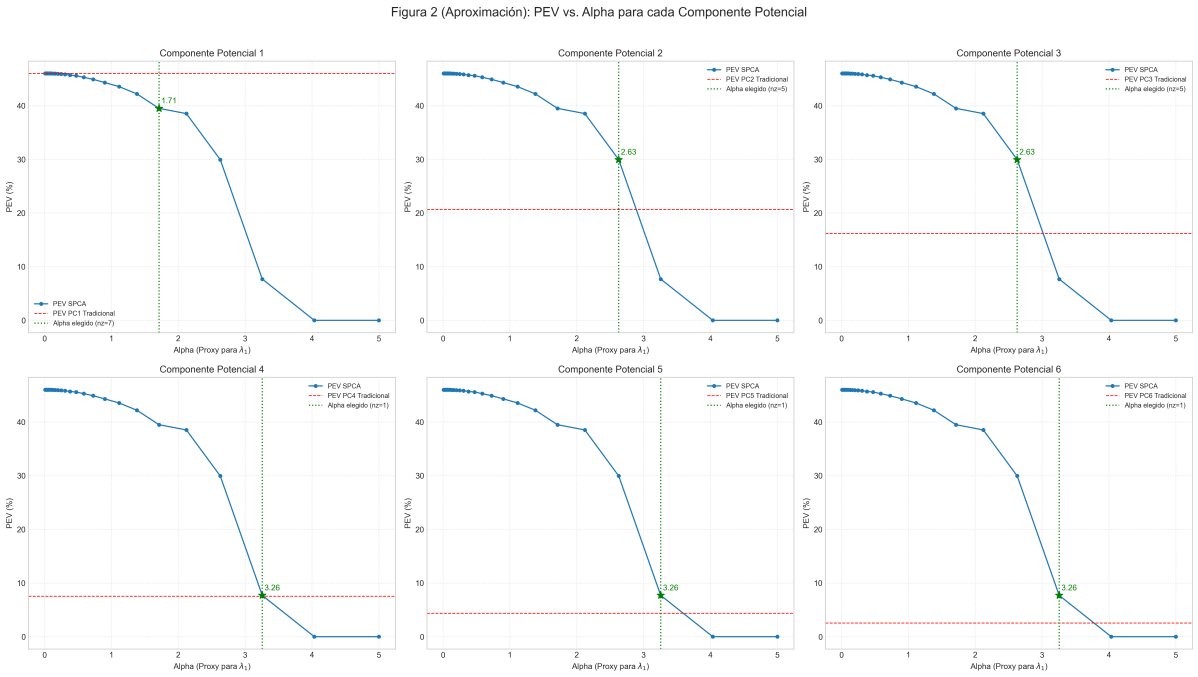

Figure 1. Aproximación de la Figura 2 del paper: PEV vs. Alpha para cada Componente Potencial. Las líneas rojas discontinuas indican el PEV del PCA tradicional. Las líneas verdes discontinuas indican el alpha elegido para cada componente para intentar igualar el número de no-ceros del paper.

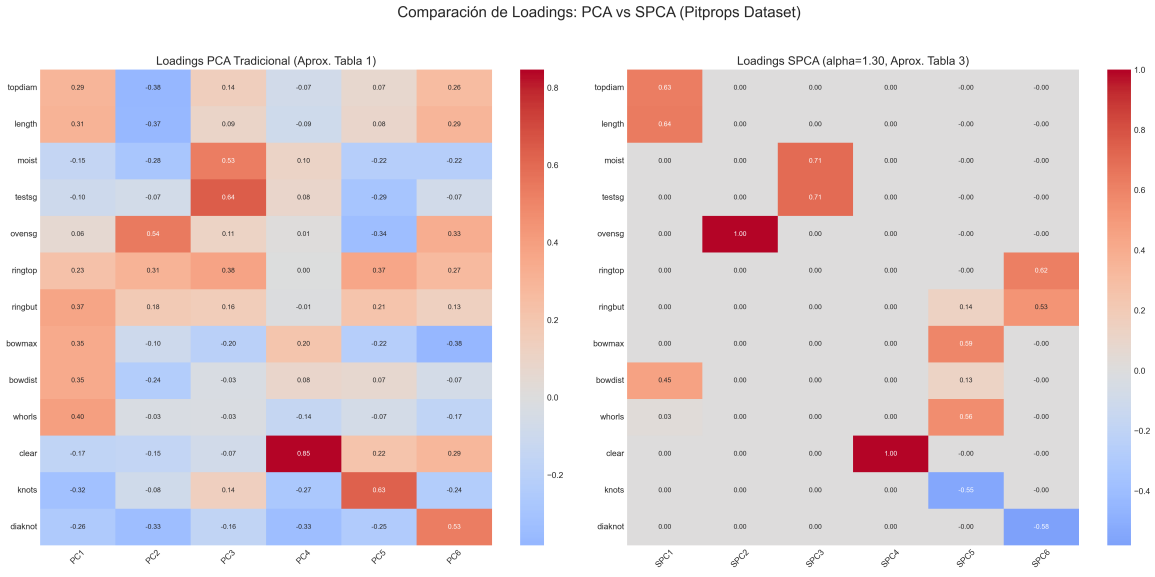

Figure 2. Comparación de Heatmaps: Loadings del PCA Tradicional (izquierda) vs. Loadings del SPCA con alpha global (derecha) - este último es una exploración inicial, no el resultado final con alpha individuales.

Visualizaciones Generadas

Conclusiones de la Replicación La replicación del ejemplo PITPROPS DATA demuestra la viabilidad de implementar SPCA utilizando herramientas estándar de Python. La clave para aproximar los resultados del paper radica en la selección cuidadosa de los parámetros de regularización (alpha en este caso) para cada componente de forma individual, buscando un equilibrio entre la varianza explicada y la dispersión deseada. Las gráficas tipo Figura 2 son esenciales para guiar esta selección. Aunque los valores exactos de varianza explicada pueden diferir debido a las variaciones en las implementaciones algorítmicas, el principio de obtener componentes más interpretables mediante la dispersión de loadings se logra con éxito. Las diferencias en la varianza ajustada acumulada sugieren que el método de scikit-learn o la estrategia de construcción secuencial sin una ortogonalización estricta pueden no capturar la misma estructura de covarianza que el método original del paper.

1. **Random Projection**
   El paper "Multivariate Technique for Detecting Variations in High-Dimensional Imagery" introduce una técnica que integra el Gráfico de Control Multivariado de Shewhart (MSCC) con métodos de proyección aleatoria (RP) para identificar cambios en la composición de células inmunes en imágenes de alta dimensionalidad.

   A continuación, se explican los puntos solicitados:

   a. **¿En qué consiste la metodología de proyecciones aleatorias?**
   La Proyección Aleatoria (RP, por sus siglas en inglés) es una técnica de reducción de dimensionalidad potente y computacionalmente eficiente. Su principio fundamental es transformar datos de alta dimensionalidad (espacio original $\mathcal{R}^p$) a un espacio de menor dimensionalidad ($\mathcal{R}^d$, donde $d \ll p$) utilizando una matriz aleatoria R de dimensiones $d \times p$.

   La transformación se realiza multiplicando la matriz de datos original X (de $n \times p$, donde n es el número de muestras y p la dimensionalidad original) por la transpuesta de la matriz de proyección aleatoria R^T (de $p \times d$), obteniendo una nueva matriz de datos proyectados Y (de $n \times d$):

   $$Y_{n\times d} = X_{n\times p} R^T_{p\times d}$$

   La idea clave es que, a pesar de la reducción drástica en dimensiones, las distancias entre los puntos en el espacio proyectado se preservan aproximadamente (con cierta distorsión controlada). Esto permite realizar análisis y aplicar algoritmos en el espacio de menor dimensión, lo que reduce la carga computacional y mitiga la "maldición de la dimensionalidad", sin perder (idealmente) la estructura esencial de los datos. El paper menciona (Sección II) que RP aproxima la distancia al cuadrado entre dos vectores $x, z \in \mathcal{R}^p$, $D^2(x, z) = \|x - z\|^2$, mediante:

   $$\hat{D}^2(x, z) = \frac{1}{d}\|Rx - Rz\|^2$$

   b. **¿Qué papel juega en este método el teorema de Johnson-Lindenstrauss?**
   El lema (o teorema) de Johnson-Lindenstrauss (JL) juega un papel fundamental como base teórica para la Proyección Aleatoria. El paper lo describe en la Sección II.A ("RP PRESERVATION LEMMA").

El lema de JL establece que un conjunto de n puntos en un espacio de alta dimensionalidad $\mathcal{R}^p$ puede ser proyectado a un espacio de dimensionalidad mucho menor $\mathcal{R}^d$ (donde d es del orden de $O(\log n/\varepsilon^2)$) tal que las distancias entre cualquier par de puntos se preservan dentro de un factor $(1 \pm \varepsilon)$, para cualquier $0 < \varepsilon < 1$. Formalmente, dada una función lineal $f : \mathcal{R}^p \to \mathcal{R}^d$, para cualquier par de puntos x, z:

$$(1-\varepsilon)\|x-z\|^2 \leq \|f(x)-f(z)\|^2 \leq (1+\varepsilon)\|x-z\|^2$$

En el contexto del método presentado:

- **Justificación teórica:** El lema de JL garantiza que tales proyecciones aleatorias que preservan distancias existen, lo que da validez al uso de RP para reducir la dimensionalidad antes de aplicar el MSCC.
- **Determinación de la dimensión reducida (d):** Aunque el lema no especifica cómo construir la matriz de proyección f (que en RP es la matriz aleatoria R), sí proporciona una guía para determinar la dimensionalidad del espacio proyectado d necesaria para lograr una cierta preservación de distancia ε para n puntos. El paper menciona (Sección IV.C) que la dimensión d se puede estimar como $d = \lceil 4\log(n)/\varepsilon^2 \rceil$ (una formulación común derivada del lema, aunque el paper cita $\lceil 4 \cdot \log(n) \rceil$ que parece omitir ε^2 o asume un ε fijo).

Así, el lema de JL asegura que la reducción de dimensionalidad mediante RP es matemáticamente sólida y que la estructura intrínseca de los datos (reflejada en las distancias euclidianas) se mantiene en gran medida.

c. **Indique los tipos de matrices aleatorias consideradas por los autores.** Los autores consideran y evalúan cuatro tipos distintos de matrices de proyección aleatoria (Sección II.B "DIFFERENT RP TYPES"):

   i. **Proyección de Achlioptas (AP):** Esta proyección introduce escasez (sparsity) en la matriz de RP, lo que la hace computacionalmente eficiente. Cada entrada r_{jl} de la matriz R se calcula según la siguiente distribución (Ecuación 4):

$$r_{jl} = \sqrt{3} \times \begin{cases} +1 & \text{con probabilidad } 1/6 \\ 0 & \text{con probabilidad } 2/3 \\ -1 & \text{con probabilidad } 1/6 \end{cases}$$

   Esto asegura que dos tercios de las entradas sean cero. El factor $\sqrt{3}$ ayuda a preservar las distancias euclidianas.

   ii. **Proyección Más-Menos Uno (PM):** Este método genera una matriz de RP densa (sin entradas cero), lo que según los autores puede permitir una mejor preservación de distancias y estructuras. Cada entrada r_{jl} se calcula independientemente según (Ecuación 5):

$$r_{jl} = \begin{cases} +1 & \text{con probabilidad } 1/2 \\ -1 & \text{con probabilidad } 1/2 \end{cases}$$

   Es computacionalmente simple y eficiente.

iii. **Proyección de Li:** Este es un enfoque alternativo para crear matrices de RP dispersas. Según el paper (página 4, Ecuación 6), cada entrada r_{jl} se calcula como:

$$r_{jl} = \begin{cases} +1 & \text{con probabilidad } 1/(2s^2) \\ 0 & \text{con probabilidad } 1 - 1/s^2 \\ -1 & \text{con probabilidad } 1/(2s^2) \end{cases}$$

donde $s = \lceil \sqrt{p} \rceil$ es un factor de escala (siendo p la dimensionalidad original). El paper indica que esto introduce escasez, aunque la descripción previa a la fórmula sugiere que la mitad de las entradas son cero, lo cual no se deduce directamente de esta fórmula para un s genérico basado en p.

iv. **Proyección Normal (NP):** También conocida como Proyección Gaussiana, utiliza entradas para la matriz de RP extraídas independientemente de una distribución normal estándar. Cada entrada r_{jl} (Ecuación 7):

$$r_{jl} \sim \mathcal{N}(0, 1)$$

Este método tiene una base estadística sólida debido a las propiedades isotrópicas de la distribución normal.

Los autores evalúan el desempeño del MSCC bajo estos diferentes tipos de proyecciones aleatorias.

2. **Usando el conjunto de datos car data, monitoree el proceso**

a. **usando cartas de control T2**

Se aplicó un monitoreo de proceso utilizando cartas de control T^2 de Hotelling sobre el conjunto de datos "cars.csv".

Metodología General

- **Carga y Preprocesamiento:** Se cargó el archivo "cars.csv", seleccionando 11 variables numéricas. Tras eliminar filas con valores NaN, se obtuvieron 111 observaciones.
- **División de Datos:** Las 111 observaciones se dividieron en una Fase I para calibración (77 observaciones, 70%) y una Fase II para monitoreo (34 observaciones, 30%).
- **Fase I - Establecimiento de Parámetros:**
  - El conjunto inicial de 77 observaciones de Fase I no presentó normalidad multivariada (test de Henze-Zirkler: HZ=1.112, p <0.001).
  - Se implementó un proceso iterativo de "limpieza" (máximo 25 iteraciones) para eliminar outliers:
    i. Se calcularon las distancias D^2 de Mahalanobis para el conjunto de datos actual (`Y_clean`).
    ii. Se estimó un umbral empírico para D^2 mediante bootstrap (1000 re-muestreos, usando el percentil 97.5).

iii. En cada iteración, se eliminó la observación con la mayor D^2 que superase dicho umbral. Se generaron gráficos diagnósticos de las D^2 y la distribución bootstrap.

- Este proceso eliminó 2 observaciones: la observación con índice original 38 ($D^2 = 26.79 >$ umbral 25.643) en la primera iteración, y la observación con índice original 42 ($D^2 = 25.53 >$ umbral 24.937) en la segunda.
- En la tercera iteración, no se detectaron más outliers. El conjunto de datos de calibración final (`Y_clean_final`) quedó con 75 observaciones.
- Este `Y_clean_final` tampoco cumplió con la normalidad multivariada (HZ=1.095, p <0.001).
- Se calcularon el vector de medias y la matriz de covarianzas inversa a partir de `Y_clean_final`.
- Se determinó un umbral final para la Fase II mediante un nuevo bootstrap sobre `Y_clean_final`, resultando en un límite de control superior (LCS) de 24.225 para el percentil 97.5.

- **Fase II - Monitoreo:**
  - Se calcularon las distancias D^2 para las 34 observaciones de la Fase II, utilizando los parámetros (media y covarianza inversa) estimados en la Fase I.
  - Estas D^2 se compararon con el LCS de 24.225. Se generó una carta de control para visualizar los resultados.

Resultados Clave (T^2)

- El proceso de Fase I estabilizó los parámetros de control tras la eliminación de 2 observaciones atípicas, resultando en un conjunto de calibración de 75 muestras.
- En la Fase II, al monitorear las 34 nuevas observaciones, **14 observaciones** (aproximadamente el 41%) fueron detectadas como fuera de control, al exceder el umbral T^2 de 24.225.
- La persistente falta de normalidad multivariada en los datos de Fase I justificó el uso de umbrales empíricos basados en bootstrap en lugar de umbrales teóricos (Chi-cuadrado).

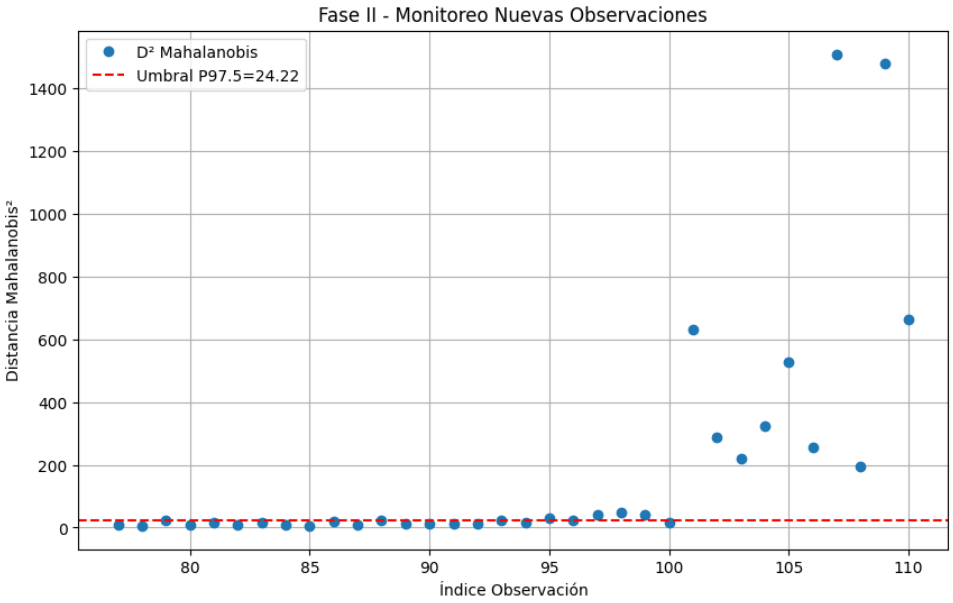

Figure 3. Carta de Control T^2 para el dataset Car Data (Fase II).

b. **usando cartas de control basadas en PCA**

Se implementó un monitoreo de proceso utilizando un enfoque basado en Análisis de Componentes Principales (PCA). Este método monitorea simultáneamente dos aspectos: la variación capturada por los componentes principales (CPs) retenidos y la variación residual o error de reconstrucción.

Metodología General

- **Carga y Preprocesamiento:** Se utilizaron los mismos datos de "cars.csv" (111 obs., 11 var. num.). Los datos fueron estandarizados (centrados y escalados a varianza unitaria) antes del PCA.
- **Fase 0 - Configuración de PCA:**
  - Se realizó un PCA sobre el conjunto de datos completo y estandarizado.
  - Se seleccionaron los CPs necesarios para explicar al menos el 90% de la varianza total. Esto resultó en la retención de **5 CPs**, que conjuntamente explicaron el 91.79% de la varianza.
  - Se calcularon los scores de estos 5 CPs y la matriz de errores.
- **División de Datos:** Tanto los scores de los 5 CPs como los datos de error se dividieron en una Fase I (77 obs.) y una Fase II (34 obs.).
- **Fase I - Establecimiento de Parámetros (Combinado):**
  - Se diseñaron dos cartas de control T^2: una para los scores de los CPs y otra para los errores.
  - Se estableció un nivel de significancia global (α_{global}) de 0.05. Mediante corrección de Bonferroni para las dos cartas, el alfa individual (α_{ind}) fue de 0.025, implicando un percentil del 97.5% para los umbrales bootstrap de cada carta.
  - Los datos iniciales de Fase I para CPs y errores no mostraron normalidad multivariada.
  - Se llevó a cabo un proceso iterativo de "limpieza" (máx. 25 iteraciones):
    i. Se calcularon D^2 y se estimaron umbrales bootstrap (P97.5) para los CPs y para los errores de forma separada.
    ii. Una observación se consideró atípica si su D^2 excedía el umbral en la carta de CPs **o** en la carta de errores.
    iii. De las observaciones candidatas, se eliminó aquella con la mayor desviación relativa (D^2/umbral) respecto a su carta correspondiente. Se generaron gráficos diagnósticos.
  - Este proceso se extendió por 16 iteraciones, eliminándose un total de 15 observaciones. Por ejemplo, en la primera iteración, se eliminó la observación con índice original 4 debido a que su D^2 en la carta de errores (17.87) superó el umbral (15.049).
  - Los conjuntos finales de calibración, `Y_clean_cps_final` (62 obs) y `Y_clean_errors_final` (62 obs), siguieron sin evidenciar normalidad multivariada.
  - Se establecieron los umbrales finales (P97.5) para la Fase II: LCS para CPs = 10.797, y LCS para Errores = 14.176.

- **Fase II - Monitoreo (Combinado):**
  - Para las 34 observaciones de Fase II, se calcularon las D^2 para los scores de CPs y para los errores, utilizando los parámetros respectivos de Fase I.
  - Una observación se clasificó como fuera de control si su D^2_{CPs} excedía 10.797 **o** su D^2_{Errores} excedía 14.176. Se generaron las cartas de control correspondientes.

Resultados Clave (PCA)

- El modelo PCA se basó en 5 componentes principales como se observa en 4.

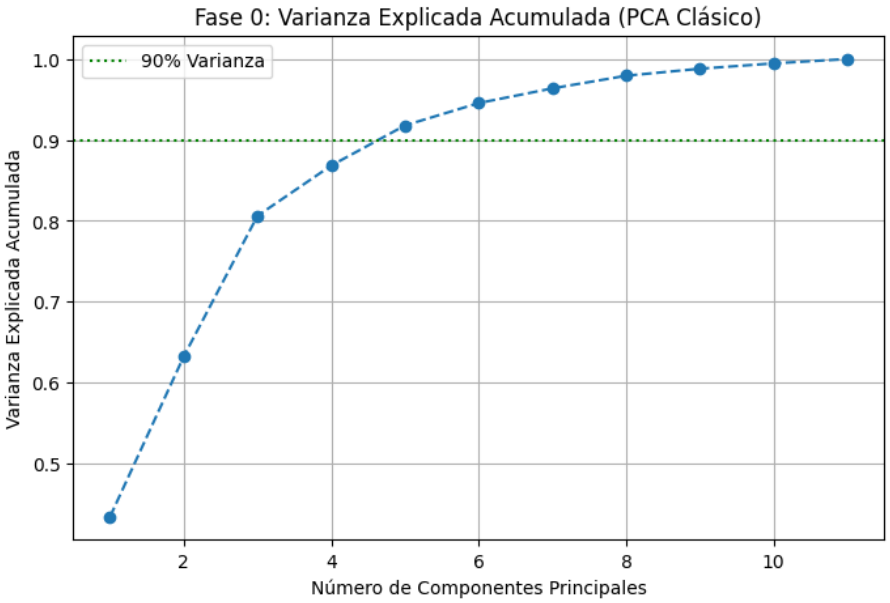

Figure 4. Decisión de la cantidad de componentes principales por la cantidad de varianza explicada.

- El proceso de calibración de Fase I requirió la eliminación de 15 observaciones para estabilizar los límites de control, resultando en conjuntos de calibración de 62 muestras.
- En la Fase II, al monitorear las 34 nuevas observaciones:
  - Algunas observaciones señalaron alarma solo en una de las cartas (e.g., la observación con índice original 82 fue fuera de control por CPs pero no por errores).
  - Otras señalaron alarma en ambas cartas (e.g., la observación con índice original 97 fue fuera de control tanto por CPs como por errores).
- En total, **19 de las 34 observaciones** (aproximadamente el 56%) de Fase II fueron detectadas como fuera de control bajo este esquema combinado de PCA.
- El enfoque PCA permitió un análisis más detallado, distinguiendo entre cambios en la estructura principal de los datos (reflejados por los CPs) y desviaciones en las correlaciones menores o ruido estructurado (reflejados por los errores de reconstrucción).

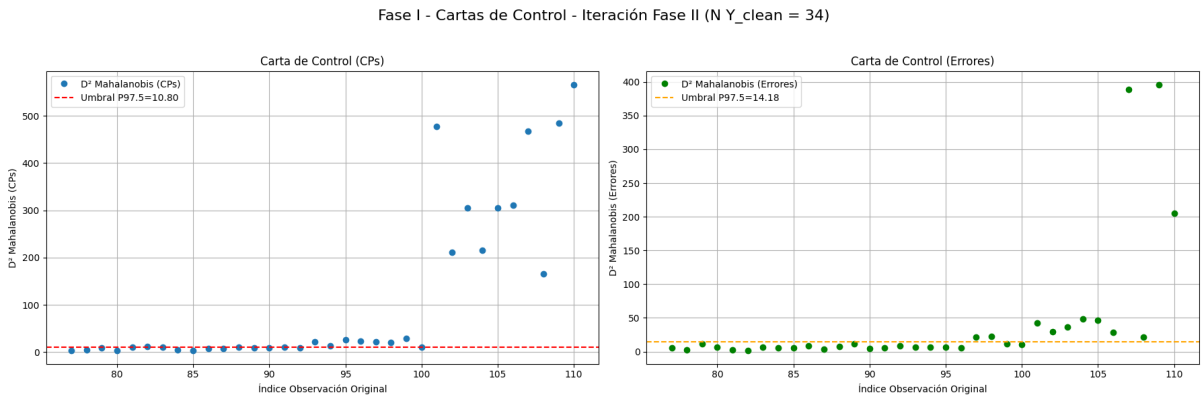

Figure 5. Cartas de Control combinadas (CPs y Errores) para el dataset Car Data (Fase II).

Fundamentos Matemáticos de los Clasificadores

A continuación se detalla la base matemática de cada uno de los cinco algoritmos de clasificación utilizados. Para todas las formulaciones, consideraremos un conjunto de datos de entrenamiento con n muestras $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, donde $\mathbf{x}_i \in \mathbb{R}^p$ es el vector de p características y $y_i \in \{1, 2, \ldots, K\}$ es la etiqueta de la clase.

(a) **Clasificador por Profundidad de Mahalanobis**

Objetivo Principal: Clasificar un nuevo punto en la clase con respecto a la cual es más "central" o "profundo". La máxima profundidad corresponde a la mínima distancia de Mahalanobis.

Supuestos: El modelo no asume una distribución de probabilidad específica, pero funciona mejor cuando las nubes de puntos de cada clase tienen una forma elipsoidal que puede ser caracterizada por una media y una matriz de covarianza.

Formulación Matemática:

i. Para cada clase $k \in \{1, \ldots, K\}$, se calculan su vector de medias $\boldsymbol{\mu}_k$ y su matriz de covarianza $\boldsymbol{\Sigma}_k$.

$$\boldsymbol{\mu}_k = \frac{1}{n_k} \sum_{\mathbf{x}_i \in \text{clase } k} \mathbf{x}_i$$

$$\boldsymbol{\Sigma}_k = \frac{1}{n_k - 1} \sum_{\mathbf{x}_i \in \text{clase } k} (\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T$$

donde n_k es el número de muestras en la clase k.

ii. La **distancia de Mahalanobis al cuadrado** de un punto $\mathbf{x}$ al centroide de la clase k se define como:

$$D_M^2(\mathbf{x}, k) = (\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k)$$

Donde $\boldsymbol{\Sigma}_k^{-1}$ es la inversa de la matriz de covarianza.

iii. La **profundidad de Mahalanobis** de un punto $\mathbf{x}$ con respecto a la clase k se define como una función monótonamente decreciente de la distancia. Una definición común es:

$$\text{Depth}_M(\mathbf{x}, k) = \frac{1}{1 + D_M^2(\mathbf{x}, k)}$$

Regla de Decisión: Se asigna un nuevo punto $\mathbf{x}_p$ a la clase k que maximiza su profundidad, lo que es equivalente a asignarlo a la clase que minimiza su distancia de Mahalanobis.

$$\hat{y} = \underset{k \in \{1,\ldots,K\}}{\arg\max} \left(\text{Depth}_M(\mathbf{x}_p, k)\right) = \underset{k \in \{1,\ldots,K\}}{\arg\min} \left(D_M^2(\mathbf{x}_p, k)\right)$$

(b) **K-Vecinos más Cercanos (K-Nearest Neighbors - KNN)**

Objetivo Principal: Clasificar un nuevo punto basándose en la clase mayoritaria de sus 'K' vecinos más cercanos en el espacio de características.

Supuestos: Es un método no paramétrico y "perezoso" (lazy learner). Su principal supuesto es que los puntos que están cerca en el espacio de características tienen una alta probabilidad de pertenecer a la misma clase.

Formulación Matemática:

i. Se define una métrica de distancia, comúnmente la **distancia Euclidiana**, $d(\mathbf{x}_i, \mathbf{x}_j)$:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{l=1}^{p}(x_{il} - x_{jl})^2} = ||\mathbf{x}_i - \mathbf{x}_j||_2$$

ii. Para un nuevo punto $\mathbf{x}_p$, se identifican los K puntos del conjunto de entrenamiento que tienen la menor distancia a $\mathbf{x}_p$. Este conjunto de vecinos se denota como $\mathcal{N}_K(\mathbf{x}_p)$.

Regla de Decisión: La clase predicha $\hat{y}$ para $\mathbf{x}_p$ es la clase más frecuente (la moda) entre las etiquetas de los vecinos en $\mathcal{N}_K(\mathbf{x}_p)$.

$$\hat{y} = \underset{k \in \{1,\dots,K\}}{\arg\max} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{N}_K(\mathbf{x}_p)} I(y_i = k)$$

donde $I(\cdot)$ es la función indicadora, que es 1 si la condición es verdadera y 0 en caso contrario.

(c) **Análisis Discriminante Lineal (LDA) - Función de Discrepancia de Fisher**

Objetivo Principal: Encontrar una proyección lineal de los datos que maximice la separación entre las medias de las clases mientras minimiza la varianza dentro de las clases.

Supuestos:

i. Los datos de cada clase k siguen una distribución Gaussiana (normal).

ii. **Supuesto clave:** Todas las clases comparten la **misma matriz de covarianza** ($\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \cdots = \mathbf{\Sigma}_K = \mathbf{\Sigma}$).

Formulación Matemática:

i. Se define la **matriz de dispersión entre clases (Between-Class Scatter Matrix)**, $\mathbf{S}_B$:

$$\mathbf{S}_B = \sum_{k=1}^{K} n_k(\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^T$$

donde $\boldsymbol{\mu}$ es la media global de todos los datos.

ii. Se define la **matriz de dispersión dentro de las clases (Within-Class Scatter Matrix)**, $\mathbf{S}_W$:

$$\mathbf{S}_W = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in \text{clase } k} (\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T$$

iii. El **criterio de Fisher** busca un vector de proyección $\mathbf{w}$ que maximice el cociente:

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}$$

Regla de Decisión: Debido a los supuestos, la frontera de decisión entre dos clases cualesquiera es lineal. La clasificación de un nuevo punto $\mathbf{x}_p$ se realiza asignándolo a la clase k que maximiza la **función discriminante lineal**:

$$\delta_k(\mathbf{x}) = \mathbf{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k - \frac{1}{2} \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k + \log(\pi_k)$$

$$\hat{y} = \arg\max_k \ \delta_k(\mathbf{x}_p)$$

donde $\boldsymbol{\Sigma}$ es la matriz de covarianza agrupada (pooled) y π_k es la probabilidad a priori de la clase k.

(d) **Clasificador de Mínima Distancia Euclidiana (Centroide más Cercano)**

Objetivo Principal: Clasificar un nuevo punto asignándolo a la clase cuyo centroide (vector medio) se encuentre a la menor distancia euclidiana.

Supuestos: Es un método paramétrico que asume que las clases son representables de forma compacta por un único centroide. Funciona mejor cuando las nubes de puntos de cada clase son esféricamente simétricas alrededor de su media y están bien separadas.

Formulación Matemática:

i. Para cada clase $k \in \{1, \ldots, K\}$, se calcula su **centroide** o vector de medias $\boldsymbol{\mu}_k$:

$$\boldsymbol{\mu}_k = \frac{1}{n_k} \sum_{\mathbf{x}_i \in \text{clase } k} \mathbf{x}_i$$

ii. Se define la **distancia Euclidiana** entre un punto $\mathbf{x}$ y el centroide de la clase k:

$$d_E(\mathbf{x}, \boldsymbol{\mu}_k) = \sqrt{\sum_{l=1}^{p} (x_l - \mu_{kl})^2} = ||\mathbf{x} - \boldsymbol{\mu}_k||_2$$

Regla de Decisión: Se asigna un nuevo punto $\mathbf{x}_p$ a la clase k que minimiza la distancia euclidiana a su centroide.

$$\hat{y} = \underset{k \in \{1,\dots,K\}}{\arg\min} \left(d_E(\mathbf{x}_p, \boldsymbol{\mu}_k)\right)$$

Esta regla de decisión define fronteras de decisión que son hiperplanos que bisecan perpendicularmente el segmento que une cada par de centroides.

(e) **Análisis Discriminante Cuadrático (QDA)**

Objetivo Principal: Extender LDA permitiendo que cada clase tenga su propia matriz de covarianza, lo que resulta en fronteras de decisión cuadráticas.

Supuestos:

i. Los datos de cada clase k siguen una distribución Gaussiana.

ii. **Supuesto clave:** Cada clase k tiene su **propia matriz de covarianza** $\boldsymbol{\Sigma}_k$. No se asume que sean iguales.

Formulación Matemática: El modelo se basa en el teorema de Bayes, buscando la clase k que maximiza la probabilidad a posteriori $P(Y = k|\mathbf{X} = \mathbf{x})$. Esto es equivalente a maximizar el logaritmo de la probabilidad conjunta $P(\mathbf{X} = \mathbf{x}|Y = k)P(Y = k)$. La función de densidad de probabilidad para la clase k es:

$$P(\mathbf{X} = \mathbf{x}|Y = k) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}_k|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)\right)$$

Regla de Decisión: Se asigna un punto $\mathbf{x}_p$ a la clase k que maximiza la **función discriminante cuadrática**, que se deriva del logaritmo de la probabilidad a posteriori (eliminando términos constantes):

$$\delta_k(\mathbf{x}) = -\frac{1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{x} - \boldsymbol{\mu}_k) + \log(\pi_k)$$

$$\hat{y} = \underset{k}{\arg\max}\ \delta_k(\mathbf{x}_p)$$

El término $(\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)$ es la distancia de Mahalanobis al cuadrado. Como este término es cuadrático en $\mathbf{x}$, las fronteras de decisión son cuadráticas.

1. Metodología y Análisis de Resultados

A continuación, se detalla la metodología empleada y se interpretan los resultados obtenidos.

Metodología Aplicada El procedimiento se dividió en dos fases principales:

(a) **Preparación de Datos:** El conjunto de datos *wine* fue cargado, dividido en entrenamiento (75%) y prueba (25%), y estandarizado (media cero, varianza unitaria).

(b) **Clasificación sobre Datos Originales:** Se evaluaron cinco modelos de clasificación: Clasificador por Profundidad de Mahalanobis, KNN, LDA, Mínima Distancia Euclidiana y QDA. Cada modelo fue entrenado y evaluado sobre los datos estandarizados (13 características).

(c) **Reducción de Dimensionalidad y Clasificación:** Se aplicó PCA a los datos de entrenamiento para seleccionar componentes que retuvieran al menos el 95% de la varianza. Los datos fueron transformados a este nuevo espacio y los cinco modelos fueron re-entrenados y re-evaluados.

(d) **Análisis Comparativo:** Los resultados de ambas fases se compilaron en una tabla para comparar el rendimiento con y sin reducción de dimensionalidad.

Análisis Descriptivo de Resultados

a. **Análisis de Componentes Principales (PCA)**
La Tabla 2 muestra la descomposición de la varianza. Se necesitan 10 componentes para superar el umbral del 95% de varianza acumulada (96.37%), reduciendo el espacio de 13 a 10 dimensiones.

Table 2. Varianza Explicada por Componente Principal

Componente	Varianza Individual	Varianza Acumulada
PC1	35.75%	35.75%
PC2	19.21%	54.96%
PC3	10.85%	65.80%
PC4	7.42%	73.22%
PC5	6.94%	80.16%
PC6	5.20%	85.36%
PC7	4.39%	89.75%
PC8	2.50%	92.25%
PC9	2.20%	94.46%
PC10	**1.92%**	**96.37%**
PC11	1.65%	98.02%
PC12	1.25%	99.27%
PC13	0.73%	100.00%

b. **Análisis Comparativo de Clasificadores**
La Tabla 3 resume el rendimiento de los modelos antes y después de aplicar PCA.

Table 3. Resumen de Rendimiento de Clasificadores

	Datos Originales (13D)		**Datos con PCA (10D)**	
Modelo	Precisión	Errores	Precisión	Errores
1) Profundidad de Mahalanobis	1.000	0	0.978	1
2) K-Nearest Neighbors (K=5)	0.933	3	0.956	2
3) Discriminante Lineal (Fisher)	0.956	2	0.978	1
4) Distancia Euclidiana Mínima	0.978	1	0.978	1
5) Discriminante Cuadrático (QDA)	1.000	0	1.000	0

Observaciones sobre los Datos Originales (13 Dimensiones):

- Dos modelos (Profundidad de Mahalanobis y QDA) alcanzan una **precisión perfecta (1.000)**.
- Mínima Distancia Euclidiana y LDA muestran un rendimiento excelente, cometiendo solo 1 y 2 errores, respectivamente.
- KNN tiene el menor rendimiento (93.3%), aunque sigue siendo alto.

Observaciones sobre los Datos con PCA (10 Dimensiones):

- **QDA es el modelo más robusto**, manteniendo una precisión perfecta de 1.000.
- El clasificador por Profundidad de Mahalanobis sufre una degradación mínima (1 error).
- **KNN y LDA mejoran su rendimiento tras PCA**, lo que sugiere un efecto de filtrado de ruido.
- **Mínima Distancia Euclidiana es perfectamente estable**, manteniendo su rendimiento con 1 error.

Conclusión Final:
Para clasificar un nuevo punto, el **Análisis Discriminante Cuadrático (QDA)** es el método más recomendable. Demostró ser el mejor modelo en el espacio original y el más robusto tras la reducción de dimensionalidad.