Técnicas de minería en bases de conocimiento

Detección de *spam* en bases textuales mediante técnicas de aprendizaje supervisado

Mónica Calzado Granados

Universidad de Granada

22 de julio de 2024

UNIVERSIDAD DE GRANADA

Gestión de los datos

- Importancia de la gestión de grandes volúmenes de datos.

Gestión de los datos

- Importancia de la gestión de grandes volúmenes de datos.
- El volumen de datos digitales crecerá hasta 175 ZB en 2025.

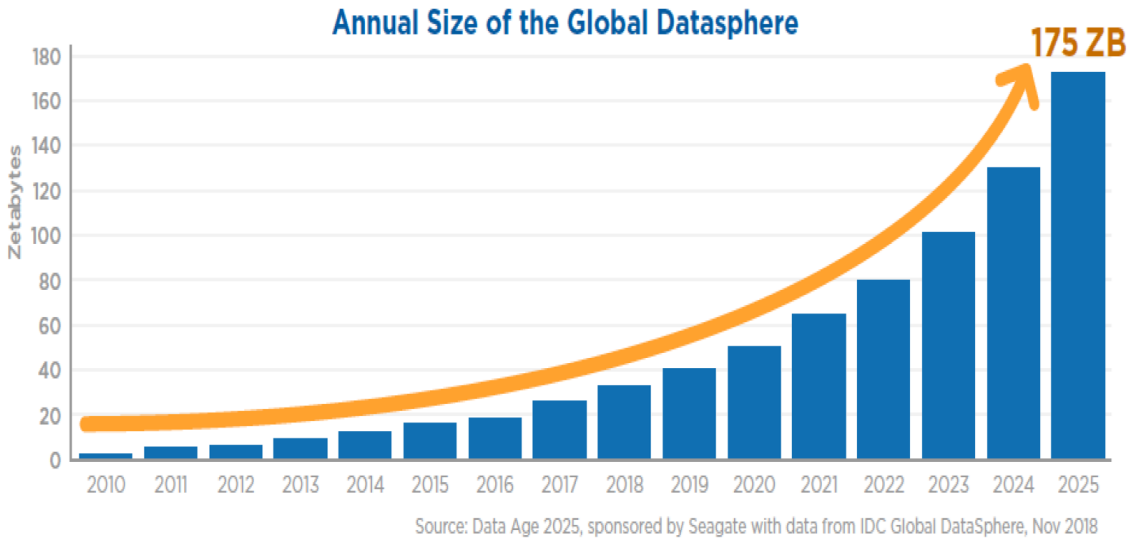


Figure 1: Imagen extraída de Forbes

Gestión de los datos

- Importancia de la gestión de grandes volúmenes de datos.
- El volumen de datos digitales crecerá hasta 175 ZB en 2025.

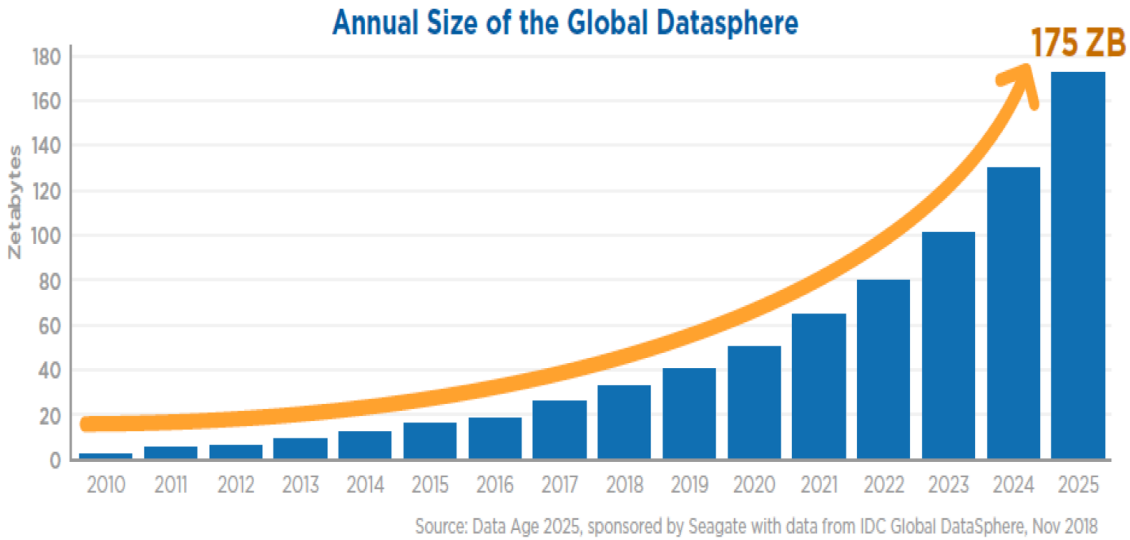


Figure 1: Imagen extraída de Forbes

- **Dos desafíos:** Almacenamiento y análisis de datos.

Motivación

- Grandes flujos de datos $\Longrightarrow$ Nuevos problemas: *spam*

Motivación

- Grandes flujos de datos $\Longrightarrow$ Nuevos problemas: *spam*
- Forma de comunicación no solicitada.

Motivación

- Grandes flujos de datos $\Longrightarrow$ Nuevos problemas: *spam*
- Forma de comunicación no solicitada.
- El 48.63% de correos electronicos fueron *spam* (Informe Kaspersky, 2022).

Motivación

- Grandes flujos de datos $\Longrightarrow$ Nuevos problemas: *spam*
- Forma de comunicación no solicitada.
- El 48.63% de correos electronicos fueron *spam* (Informe Kaspersky, 2022).
- Es fundamental crear métodos de detección de *spam*.

Objetivos

- Proporcionar una visión integral del proceso de **Minería de Textos**.
- Explorar las bases teóricas de varias técnicas de minería.
- Buscar un ejemplo de aplicación real.
- Demostrar la utilidad de las técnicas empleadas en el campo de la Minería de Textos.

KDD vs KDT

- **Knowledge Discovery from Databases** (KDD)

  El proceso no trivial de identificar patrones válidos, novedosos, potencialmente útiles y, en última instancia, comprensibles a partir de los datos (Fayyad et al., 1996).

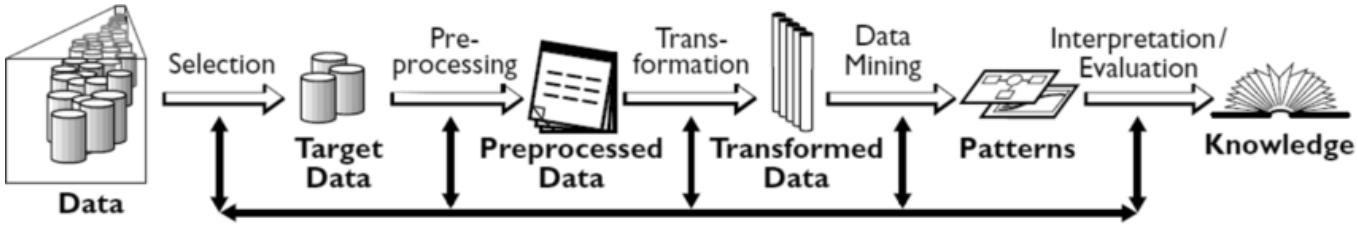


Figure 2: Fases del KDD (Fayyad et al., 1996)

KDD vs KDT

- **Knowledge Discovery from Databases** (KDD)

- **Knowledge Discovery from Text** (KDT)
  - Principal diferencia con KDD: preprocesado más complejo

KDD vs KDT

- **Knowledge Discovery from Databases** (KDD)

- **Knowledge Discovery from Text** (KDT)
  - Principal diferencia con KDD: preprocesado más complejo
  - Técnicas de PLN: tokenización, etiquetado, stemming, eliminar stop words...

KDD vs KDT

- **Knowledge Discovery from Databases** (KDD)

- **Knowledge Discovery from Text** (KDT)
  - Principal diferencia con KDD: preprocesado más complejo
  - Técnicas de PLN: tokenización, etiquetado, stemming, eliminar stop words...
  - Problema: pérdida de contexto

KDD vs KDT

- **Knowledge Discovery from Databases** (KDD)

- **Knowledge Discovery from Text** (KDT)
  - Principal diferencia con KDD: preprocesado más complejo
  - Técnicas de PLN: tokenización, etiquetado, stemming, eliminar stop words...
  - Problema: pérdida de contexto
  - Sistemas de representación de conocimiento: redes semánticas, reglas de producción, ontologías...

Naive Bayes

Problema de clasificación con n atributos $X = \{x_1, x_2, \ldots, x_n\}$ y k clases $c_i \in \Omega = \{c_1, c_2, \ldots, c_k\}$.

Naive Bayes

Problema de clasificación con n atributos $X = \{x_1, x_2, \ldots, x_n\}$ y k clases $c_i \in \Omega = \{c_1, c_2, \ldots, c_k\}$.

- **Teorema de Bayes** aplicado a clasificación:

$$P(c_i|x_1, ..., x_n) = \frac{P(c_i) \cdot P(x_1, ..., x_n|c_i)}{P(x_1, ..., x_n)}$$

Naive Bayes

Problema de clasificación con n atributos $X = \{x_1, x_2, \ldots, x_n\}$ y k clases $c_i \in \Omega = \{c_1, c_2, \ldots, c_k\}$.

- **Teorema de Bayes** aplicado a clasificación:

$$P(c_i|x_1, ..., x_n) = \frac{P(c_i) \cdot P(x_1, ..., x_n|c_i)}{P(x_1, ..., x_n)}$$

- **Hipótesis MAP**:

$$C_{\text{MAP}} = \arg \max_{c_i \in \Omega} P(c_i \mid x_1, ..., x_n)$$

Naive Bayes

Problema de clasificación con n atributos $X = \{x_1, x_2, \ldots, x_n\}$ y k clases $c_i \in \Omega = \{c_1, c_2, \ldots, c_k\}$.

- **Teorema de Bayes** aplicado a clasificación:

$$P(c_i|x_1, ..., x_n) = \frac{P(c_i) \cdot P(x_1, ..., x_n|c_i)}{P(x_1, ..., x_n)}$$

- **Hipótesis MAP**:

$$C_{\text{MAP}} = \arg\max_{c_i \in \Omega} P(c_i \mid x_1, ..., x_n)$$

- **Problema:** Calcular $P(x_1, ..., x_n|c_i)$ es computacionalmente insostenible.

Naive Bayes

Problema de clasificación con n atributos $X = \{x_1, x_2, \ldots, x_n\}$ y k clases $c_i \in \Omega = \{c_1, c_2, \ldots, c_k\}$.

- **Teorema de Bayes** aplicado a clasificación:

$$P(c_i|x_1, ..., x_n) = \frac{P(c_i) \cdot P(x_1, ..., x_n|c_i)}{P(x_1, ..., x_n)}$$

- **Hipótesis MAP**:

$$C_{\text{MAP}} = \arg\max_{c_i \in \Omega} P(c_i \mid x_1, ..., x_n)$$

- **Problema:** Calcular $P(x_1, ..., x_n|c_i)$ es computacionalmente insostenible.
- **Solución:** Asumir independencia.

$$P(x_1, ..., x_n|c_i) = \prod_{x_j \in X} P(x_j|c_i)$$

Naive Bayes

Problema de clasificación con n atributos $X = \{x_1, x_2, \ldots, x_n\}$ y k clases $c_i \in \Omega = \{c_1, c_2, \ldots, c_k\}$.

- **Teorema de Bayes** aplicado a clasificación:

$$P(c_i|x_1, ..., x_n) = \frac{P(c_i) \cdot P(x_1, ..., x_n|c_i)}{P(x_1, ..., x_n)} = \frac{P(c_i) \cdot \prod_{x_j \in X} P(x_j|c_i)}{P(x_1, ..., x_n)}$$

Naive Bayes

Problema de clasificación con n atributos $X = \{x_1, x_2, \ldots, x_n\}$ y k clases $c_i \in \Omega = \{c_1, c_2, \ldots, c_k\}$.

- **Teorema de Bayes** aplicado a clasificación:

$$P(c_i|x_1, ..., x_n) = \frac{P(c_i) \cdot P(x_1, ..., x_n|c_i)}{P(x_1, ..., x_n)} = \frac{P(c_i) \cdot \prod_{x_j \in X} P(x_j|c_i)}{P(x_1, ..., x_n)}$$

- **Hipótesis MAP**:

$$C_{\text{MAP}} = \arg\max_{c_i \in \Omega} P(c_i \mid x_1, ..., x_n)$$

$$= \arg\max_{c_i \in \Omega} P(c_i) \prod_{j=1}^{n} P(x_j \mid c_i)$$

Tipos de clasificadores Naive Bayes

Dependiendo de la distribución de $P(x_i \mid y)$:

Tipos de clasificadores Naive Bayes

Dependiendo de la distribución de $P(x_i \mid y)$:

- **NB Multinomial**
  - Variables de entrada multivariantes.
  - Conteo de frecuencia relativa:

$$P(x_i \mid y) = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$

Tipos de clasificadores Naive Bayes

Dependiendo de la distribución de $P(x_i \mid y)$:

- **NB Multinomial**
- **NB Gaussiano**
  - Variables de entrada continuas.
  - Función de densidad de probabilidad gaussiana:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

Tipos de clasificadores Naive Bayes

Dependiendo de la distribución de $P(x_i \mid y)$:

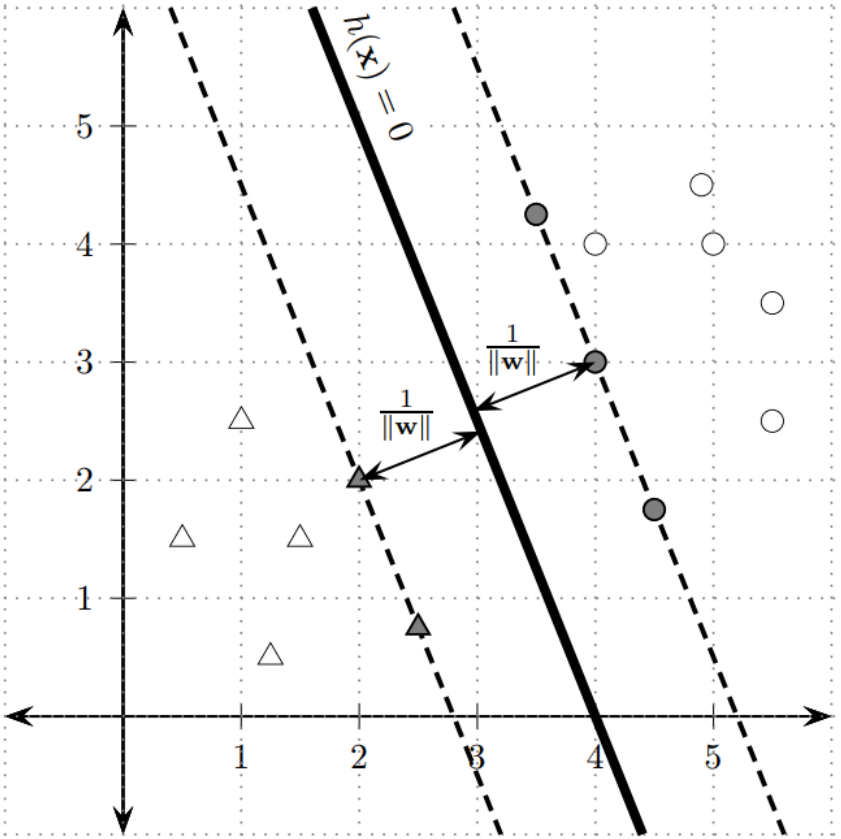
- **NB Multinomial**
- **NB Gaussiano**
- **NB de Bernoulli**
  - Variables de entrada binarias
  - Función de probabilidad de Bernoulli:

$$P(x_i \mid y) = P(x_i = 1 \mid y)x_i + (1 - P(x_i = 1 \mid y))(1 - x_i)$$

Problema lineal

- Hiperplano de separación lineal:

$$h(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b = 0$$



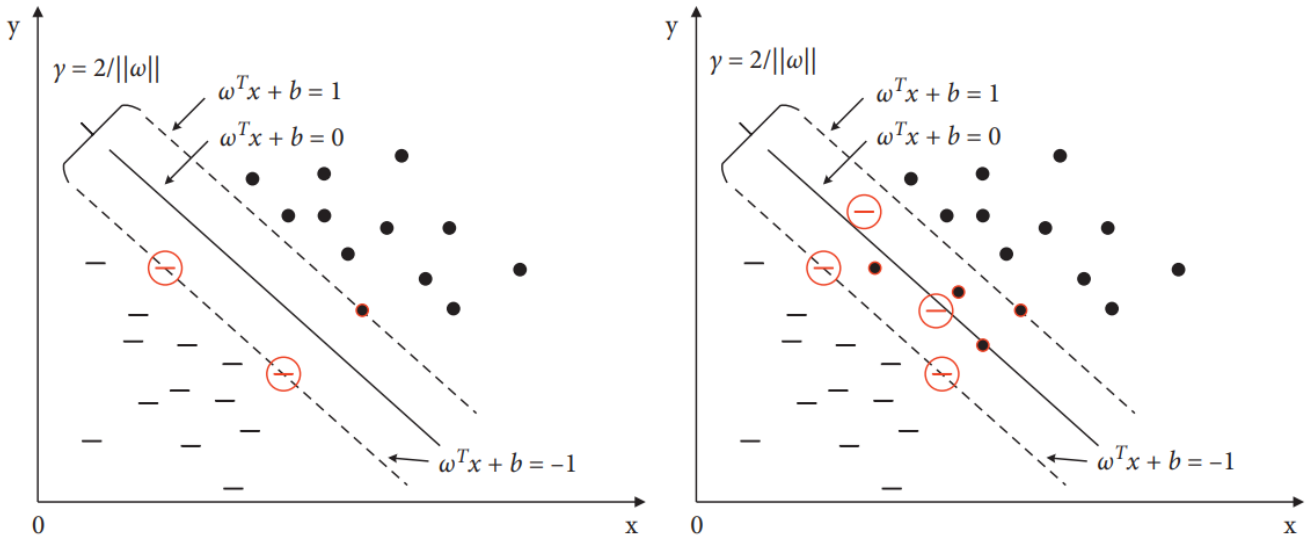
Problema lineal

Margen duro

- **Función objetivo:** $\min_{\mathbf{w},b} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 \right\}$
- **Restricciones lineales:** $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \quad \forall \mathbf{x}_i \in \mathbf{D}.$

Margen blando

- **Función objetivo:** $\min_{\mathbf{w},b,\xi_i} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{n} \xi_i \right\}$
- **Restricciones lineales:** $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \forall \mathbf{x}_i \in \mathbf{D}, \ \xi_i \geq 0.$



Problema dual asociado

Margen duro

- **Función objetivo:** $\max_{\boldsymbol{\alpha}} \left\{ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \right\}$
- **Restricciones lineales:** $\sum_{i=1}^{n} \alpha_i y_i = 0,$
  $\alpha_i \geq 0, \quad \forall i = 1, ..., n.$

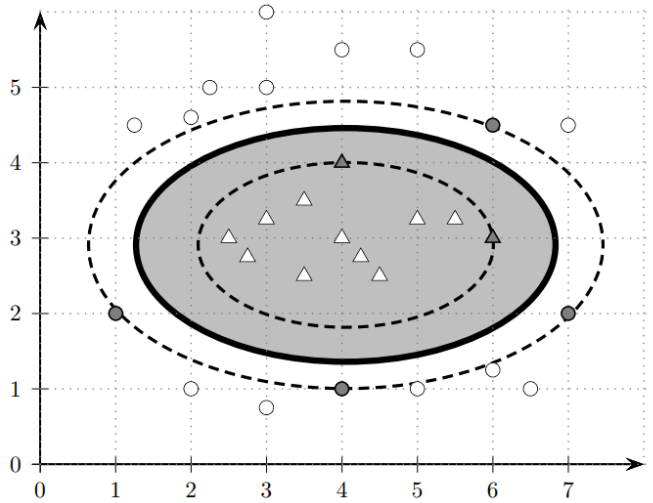
Margen blando

- **Función objetivo:** $\max_{\boldsymbol{\alpha}} \left\{ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \right\}$
- **Restricciones lineales:** $\sum_{i=1}^{n} \alpha_i y_i = 0,$
  $0 \leq \alpha_i \leq C, \quad \forall i = 1, ..., n.$

Problema NO lineal

- Nuevo hiperplano de separación lineal:

$$h(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b = 0$$



Nuevo problema dual

- **Función objetivo:**
  $\max_{\boldsymbol{\alpha}} \left\{ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) \right\}$
- **Restricciones lineales:** $\sum_{i=1}^{n} \alpha_i y_i = 0,$
  $0 \leq \alpha_i \leq C, \quad \forall i = 1, ..., n.$

Truco del kernel

- Es costoso hallar ϕ y calcular $\phi(\mathbf{x}_i)$, $\forall \mathbf{x}_i \in \mathbf{D}$.
- **Función kernel:** basta con calcular $\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$.

Ejemplos de funciones kernel

- **Kernel lineal**:

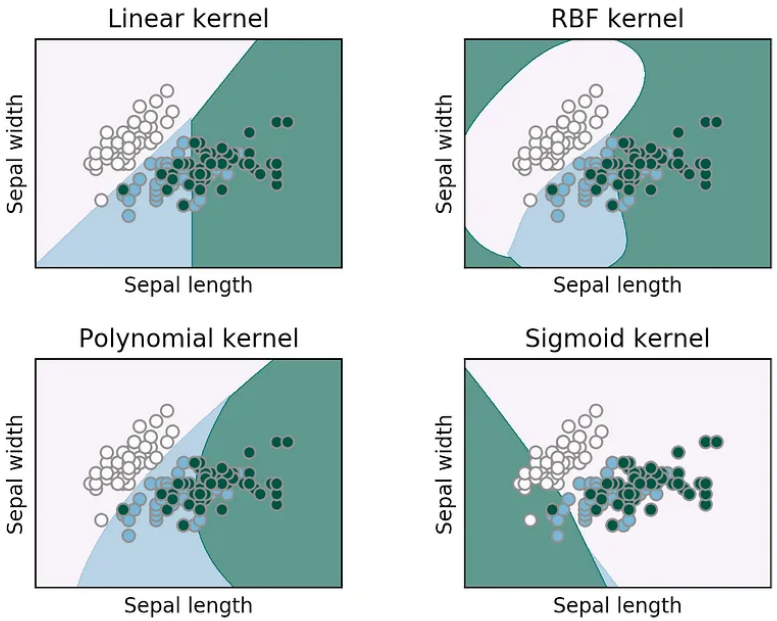
$$k(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y}$$

- **Kernel polinómico**:

$$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y} + c)^n$$

- **Kernel gaussiano**:

$$k(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2)$$



Definición del problema

- **Lenguaje de programación:** Python. Scikit-Learn, Pandas y NLTK.
- **Entorno**: Jupyter Notebook
- Conjuntos de datos extraídos de *Kaggle*.
  - Temas corporativos, académicos, foros...

- Problema de clasificación binaria: Spam y Ham

Visualización y análisis exploratorio (1)

- Limpieza de datos.
- Representación de los datos: tokenización.

Label	**Texto original**	**Texto tokenizado**
Spam	Subject: make $3500 per week using your home computer! put my free software in your computer... start making huge amounts of cash... without working!!! http://www.adclick.ws/p.cfm?o=315&s=pk007	['make', 'week', 'using', 'home', 'computer', 'put', 'free', 'software', 'computer', 'start', 'making', 'huge', 'amounts', 'cash', 'working', 'URL']

Table 1: Ejemplo de correo antes y después de la limpieza y tokenización

Visualización y análisis exploratorio (2)

- **Distribución de los datos**

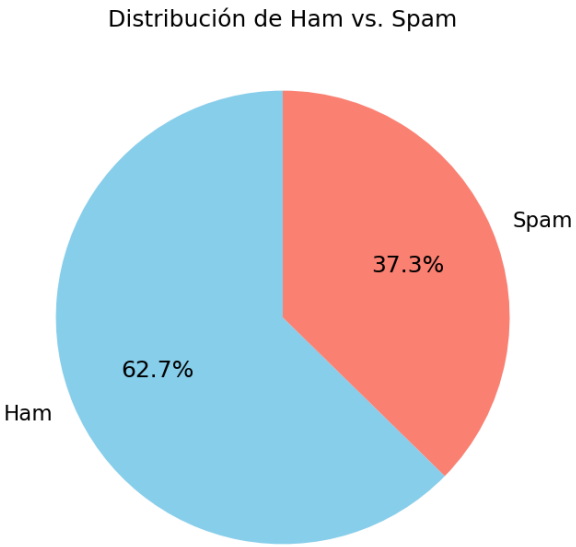


Figure 4: Desbalanceo de clases

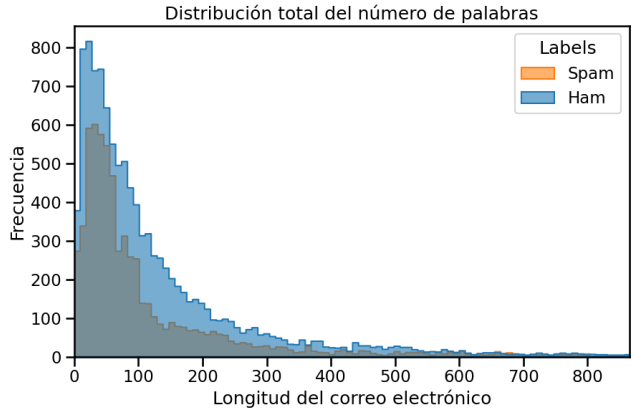


Figure 5: Frecuencia total de palabras

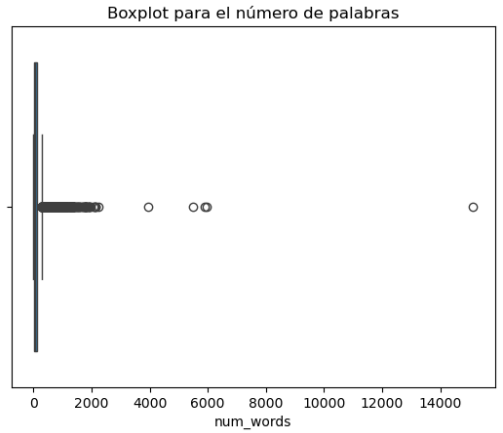


Figure 6: Boxplot del número de palabras

Visualización y análisis exploratorio (3)

- **Análisis del contenido de los datos:** word clouds y n-gramas

Figure 7: Comparación de word clouds

Preprocesado (1)

- Selección de instancias
- Selección de características
- Balanceo de clases

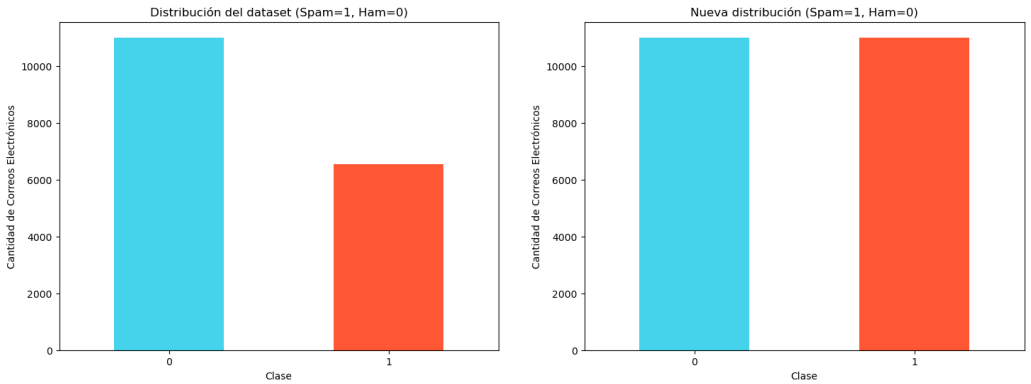


Figure 8: Aplicando balanceo de clases

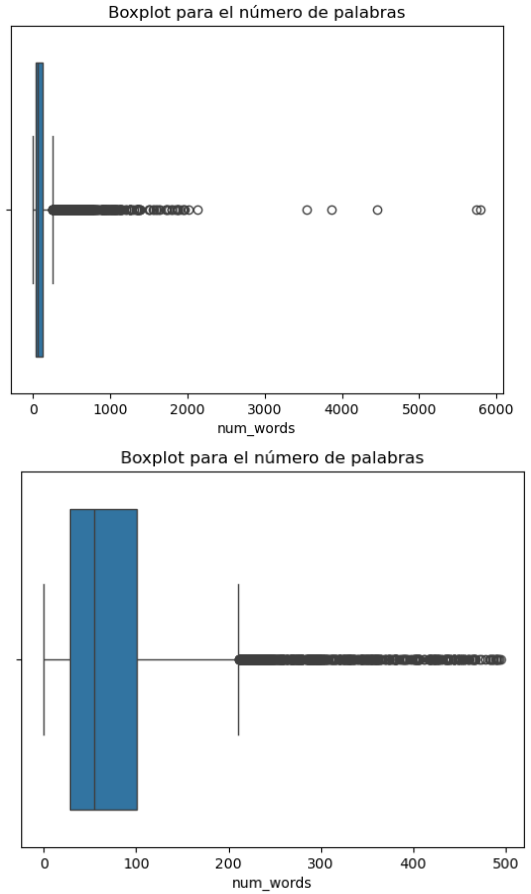


Figure 9: Boxplot del número de palabras

Preprocesado (2)

- Representación de los datos: Vectorización
  - Conteo de palabras
  - Ponderación TF-IDF (Frecuencia de Término - Inversa de la Frecuencia del Documento)

Algoritmos elegidos

- **Naive Bayes Multinomial:** número de apariciones de cada término. Complejidad $O(n \cdot m)$.

Algoritmos elegidos

- **Naive Bayes Multinomial:** número de apariciones de cada término. Complejidad $O(n \cdot m)$.
- **Naive Bayes de Bernoulli:** aparición o no de cada término. Complejidad $O(n \cdot m)$.

Algoritmos elegidos

- **Naive Bayes Multinomial:** número de apariciones de cada término. Complejidad $O(n \cdot m)$.
- **Naive Bayes de Bernoulli:** aparición o no de cada término. Complejidad $O(n \cdot m)$.
- **SVC:** permite elegir distintos kernel. Complejidad $O(n^2 \cdot m)$ a $O(n^3 \cdot m)$.

Algoritmos elegidos

- **Naive Bayes Multinomial:** número de apariciones de cada término. Complejidad $O(n \cdot m)$.
- **Naive Bayes de Bernoulli:** aparición o no de cada término. Complejidad $O(n \cdot m)$.
- **SVC:** permite elegir distintos kernel. Complejidad $O(n^2 \cdot m)$ a $O(n^3 \cdot m)$.
- **LinearSVC:** enfoque lineal. Complejidad $O(n \cdot m)$.

Algoritmos elegidos

- **Naive Bayes Multinomial:** número de apariciones de cada término. Complejidad $O(n \cdot m)$.
- **Naive Bayes de Bernoulli:** aparición o no de cada término. Complejidad $O(n \cdot m)$.
- **SVC:** permite elegir distintos kernel. Complejidad $O(n^2 \cdot m)$ a $O(n^3 \cdot m)$.
- **LinearSVC:** enfoque lineal. Complejidad $O(n \cdot m)$.
- **SGDClassifier:** enfoque lineal con aprendizaje estocástico por descenso de gradiente. Complejidad $O(n \cdot m)$.

Resultados de la clasificación (1)

- Holdout + validación cruzada estratificada.
- Métricas obtenidas en fase de prueba:

Clasificador	Accuracy	Precision	Recall	F1-Score	AUC
NB Multinomial (Count)	0.974114	0.974112	0.974173	0.974114	0.993202
NB Multinomial (TF-IDF)	0.983651	0.983641	0.983659	0.983649	0.998555
NB Bernoulli (Count)	0.945958	0.948644	0.946421	0.945910	0.994269
SVC (Count)	0.980018	0.980193	0.980160	0.980018	0.995960
SVC (TF-IDF)	0.992507	0.992503	0.992508	0.992506	0.999371
LinearSVC (Count)	0.987738	0.987770	0.987825	0.987738	0.997652
LinearSVC (TF-IDF)	0.988647	0.988655	0.988717	0.988646	0.999242
SGDClassifier (Count)	0.988193	0.988182	0.988243	0.988192	0.996938
SGDClassifier (TF-IDF)	0.989555	0.989540	0.989595	0.989554	0.999161

Resultados de la clasificación (2)

- La vectorización TF-IDF obtiene mejores resultados.

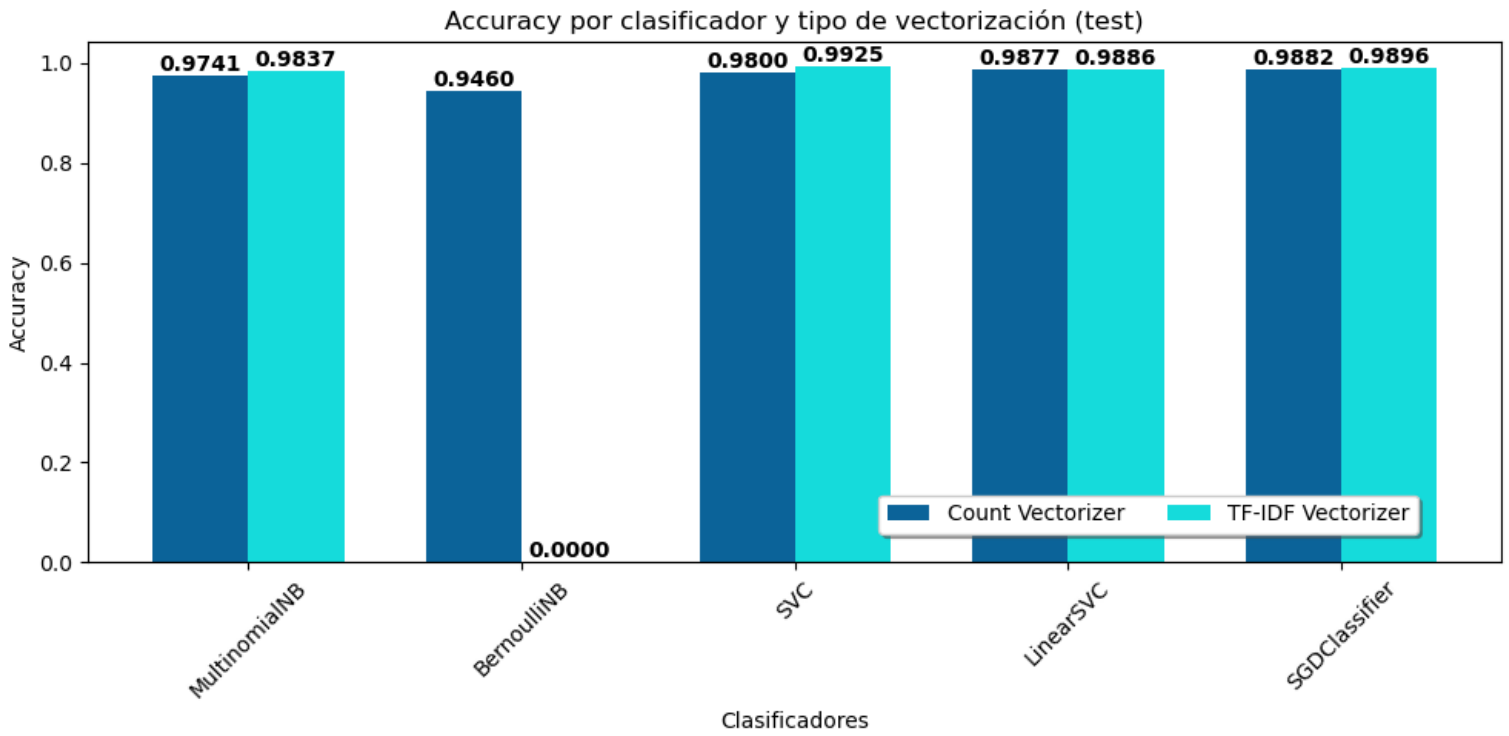


Figure 10: Comparación de accuracy en test

Resultados de la clasificación (3)

- Naive Bayes Bernoulli fue el algoritmo con peor desempeño.

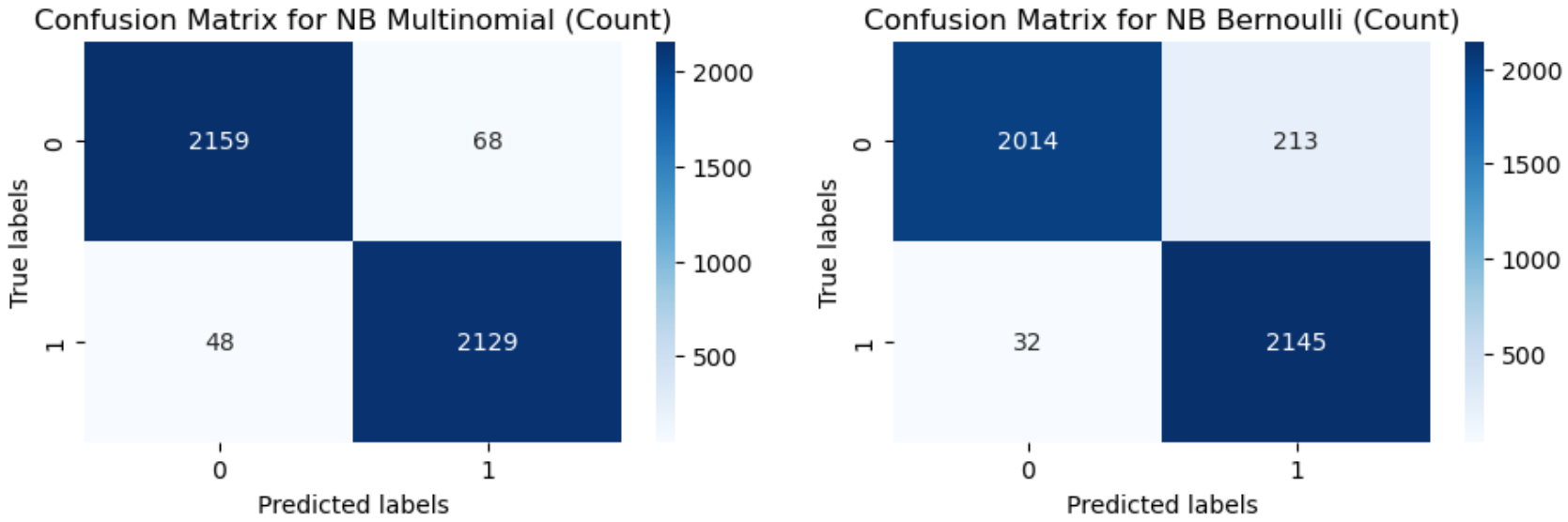


Figure 11: Comparación de matrices de confusión para test

Resultados de la clasificación (4)

El área bajo la curva ROC es casi el total para todos los modelos:

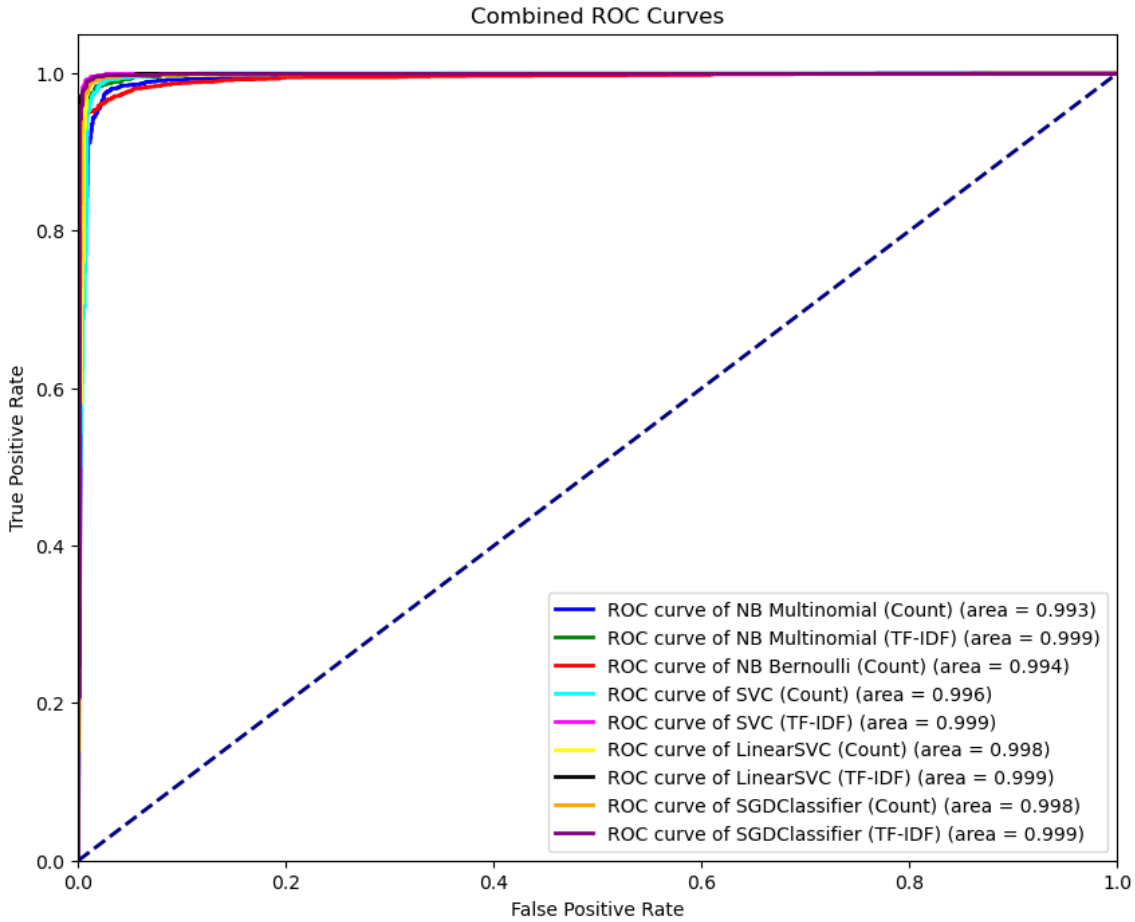


Figure 12: Comparación de curvas ROC

Otros enfoques: redes semánticas (1)

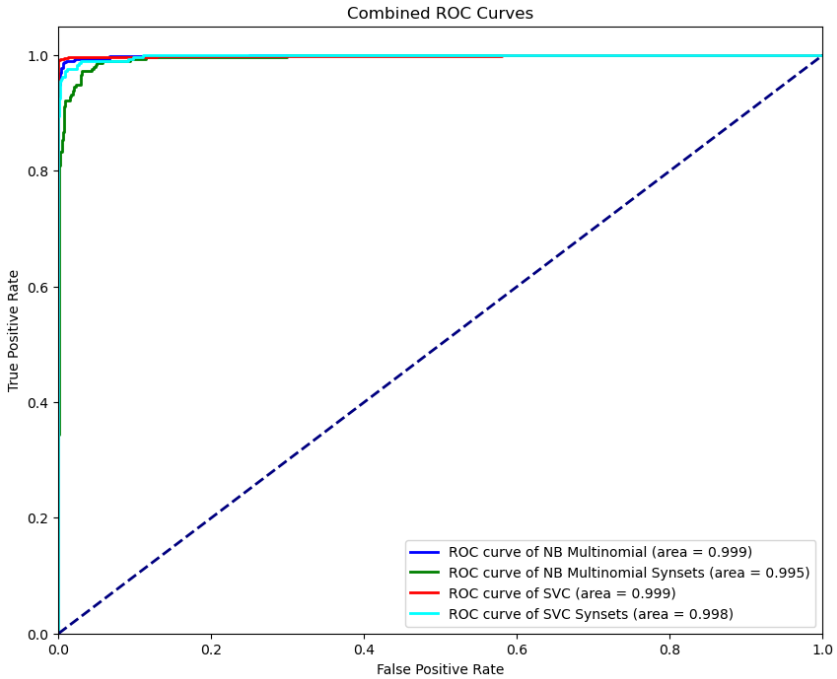
- Obtener synsets e hiperónimos de un término.
- Ejemplo: 'dollar' y 'euro' son hipónimos de 'monetary_unit'.

Label	**Texto original**	**Texto tokenizado**
Spam	fight risk cancer URL slim guaranteed lose lbs days URL	['military_action', 'danger', 'malignant_tumor', 'address', 'change_state', 'pledge', 'lose', 'avoirdupois_unit', 'time_unit', 'address']

Table 2: Ejemplo de correo antes y después de tokenización mediante synsets

Otros enfoques: redes semánticas (2)

- Resultados ligeramente peores a la tokenización normal.



Clasificador	Accuracy	Precision	Recall	F1-Score	AUC
NB Multinomial	0.990421	0.990341	0.990482	0.990408	0.999061
NB Multinomial Synsets	0.970727	0.970919	0.955656	0.962917	0.994958
SVC	0.993614	0.993604	0.993604	0.993604	0.998941
SVC Synsets	0.983947	0.982477	0.977371	0.979883	0.998355

Table 3: Métricas en fase de prueba (SpamAssassin)

Conclusiones

- Es viable la detección de *spam* mediante técnicas de minería.

Conclusiones

- Es viable la detección de *spam* mediante técnicas de minería.
- Se han encontrado características y patrones distintivos.

Conclusiones

- Es viable la detección de *spam* mediante técnicas de minería.
- Se han encontrado características y patrones distintivos.
- Dificultades:
  - Procesar texto
  - Grandes volúmenes de datos $\Longrightarrow$ sobrecarga computacional

Conclusiones

- Es viable la detección de *spam* mediante técnicas de minería.
- Se han encontrado características y patrones distintivos.
- Dificultades:
  - Procesar texto
  - Grandes volúmenes de datos $\Longrightarrow$ sobrecarga computacional
- Resultados inmejorables: métricas por encima del 95%.

Conclusiones

- Es viable la detección de *spam* mediante técnicas de minería.
- Se han encontrado características y patrones distintivos.
- Dificultades:
  - Procesar texto
  - Grandes volúmenes de datos $\Longrightarrow$ sobrecarga computacional
- Resultados inmejorables: métricas por encima del 95%.
- Otros enfoques no han conseguido mejorar los resultados.

Trabajos Futuros

- Análisis de sentimientos.

Trabajos Futuros

- Análisis de sentimientos.
- Análisis de las URLs (detección de phishing).

Trabajos Futuros

- Análisis de sentimientos.
- Análisis de las URLs (detección de phishing).
- Uso de algoritmos más explicativos.

Fin de la Presentación

Gracias por su atención.