CO_2 classification analysis for AI models

Juan Felipe Gallo Rendón* *Engineering department*
University of Antioquia
Medellín, Colombia

***Abstract*—**

***Index Terms*—Cluster analysis, K-means, Classification models,**

I. Introduction

II. Methodology

Classification methods are applied to the selected dataset. A workflow comprising six procedures is followed: **univariate analysis**, **cluster analysis**, **k-means clustering**, **linear classification**, and **nonlinear classification**. Upon completion of these steps, the results are compared and the best-performing model is identified.

A. Univariate Analysis

In this stage, the dataset **HFCO2.csv**—originating from a CO_2-emissions report on the Hugging Face platform—is profiled to characterize marginal distributions, identify outliers, and screen variables for subsequent modeling. The variables under study are enumerated below. Variables such as `modelId`, `datasets`, `co2_reported`, `createdat`, and `libraryname` are excluded due to limited analytical relevance; `database_efficiency` is removed because it is functionally dependent on `co2_eq_emissions` and `size`. Because no native target is present, classification labels are defined to enable supervised evaluation. First, `model_type` is created to align the `performance_score` with the appropriate metric family (e.g., accuracy, F1, Rouge); rows lacking performance metrics are discarded. Univariate summaries and frequency counts are then computed for qualitative variables (e.g., `model_type`) and proportions are reported for `domain`; variables exhibiting near-constant distributions (e.g., `library_name` = `pytorch`), or extreme sparsity (`geographical_location`, `environment`) are removed. Finally, a fairness-oriented label is introduced to relate predictive performance to environmental impact, yielding four categories—*Fair and Efficient*, *Powerful but Expensive*, *Green but Weak*, and *Inefficient*—and a derived Boolean variable `is_fair`. The resulting curated dataset and label structure provide the basis for the next methodological step, cluster analysis.

B. Cluster analysis

A hierarchical clustering analysis was applied to segment AI models using numeric performance-and-cost variables. Records with N/A/∞ were removed and features were were standardized to z-scores, to mitigate dominance of high skewed counts in features like `downloads` and `likes`, simple transformation (log/winsorzing) were considered. Two distances were evaluated: Euclidean and Mahalanobis, the latter computed with a Ledoit-Wold covariance estimator. Four linkage rules (Ward, average, complete and single) were compared. The dendogram was cut either at a target k (k $\in \{3, 4\}$) to improve size balance or at the k that maximized the average silhouette; the equivalent distance threshold reproducing that partition was also recorded. Internal quality was assessed with the silhouette score, while cophenetic correlation was used descriptively to gauge dendogram fidelity. As external validation, labels (`is_fair`, `fairness_class`, `model_type`) were not used for training and were cotrasted post-hoc via contingency tables, χ^2/Crammer's V, and adjusted Rand index. Cluster interpretation relied on profiles in original units and on z-centroids; a |z| ranking highlighted the most discriminative variables. A comparative summary across configurations (metrics, thresholds, balance) and visualizations (dedongrams with horizontal cut, stacked proportions and heatmaps) were produced for review.Final selection prioritized overall separability and operational usefulness, balancing silhouette with stability, cluster-size balance and consistency with external labels.

K-means analysis: K-means was applied to the same numeric variables used in the hierarchical analysis (*performance_score*, *co2_eq_emissions*, *likes*, *downloads*, *size*). Rows with non-numeric/inf values were removed and features were standardized to z-scores (`StandardScaler`). We explored $k \in \{2, \ldots, 10\}$ with `k-means++` initialization ($n\_init = 50$, `max_iter`=500, `random_state`=42).

For each k we reported: Silhouette (higher is better), Calinski–Harabasz (CH; higher is better), Davies–Bouldin (DBI; lower is better), and the inertia (*elbow* curve). The primary selector was the maximum Silhouette; CH/DBI and the elbow were used as secondary evidence.

For the selected k we exported cluster sizes, original-unit profiles (count/mean/median), z-centroids, and a variable ranking using $\max|z|$ (to highlight the most discriminative features). External labels (*is_fair*, *clasification_fairness*, *model_type*) were held out from training and only used for validation via contingency tables, χ^2/Cramér's V, Fisher's exact test when 2×2, and ARI when applicable.

C. Multivariate Normality & LDA Feasibility — Methods

Aim. Check whether the feature vector X=`performance_score`, `co2_eq_emissions`, `likes`, `downloads`, `size` is (approximately) multivariate normal—globally and within classes—so that classical LDA is justified.

Preprocessing. Remove rows with NA/inf; standardize all variables to z–scores.

Robust scatter & distances. Fit a shrinkage (Ledoit–Wolf) covariance $\widehat{\Sigma}$ to mitigate outliers. Compute squared Mahalanobis distances $d_i^2 = (x_i - \widehat{\mu})^\top \widehat{\Sigma}^{-1}(x_i - \widehat{\mu})$ (i) globally and (ii) per class.

Normality diagnostic. Under MVN with p variables, $d_i^2 \sim \chi_p^2$. Compare empirical quantiles of $\{d_i^2\}$ against χ_p^2 using QQ–plots: global and class–conditional. Large, systematic upward deviations (heavy tails/mixtures) indicate non–normality.

Optional formal checks. (a) Mardia or Henze–Zirkler tests on z–scores; (b) Box's M (or visual comparison) for equality of class covariances.

Decision rule. nosep,leftmargin=*

1) *Proceed with classical LDA* if QQ–plots track the χ_p^2 line reasonably well (globally *and* by class) and class covariances appear similar.
2) *Prefer robust/regularized LDA or nonparametric classifiers* (e.g., shrinkage LDA, logistic regression, tree–based) if tails inflate, classes differ in scatter, or tests reject MVN/homoscedasticity.

III. Results and discussions

LOREMPIPSUM

A. Univariate Analysis

The database **HFCO2.csv** is a product of the CO_2 emissions report generated on the Hugging Face platform. In this case, the analysis is focused on the following variables:

1) `co2_eq_emissions`: the resulting carbon footprint
2) `downloads`: number of model downloads
3) `likes`: number of model likes
4) `performance_metrics`: (accuracy, F1, Rouge-1, Rouge-L)
5) `performance_score`: the harmonic mean of the normalized performance metrics
6) `size`: size of the final trained model in MB

Variables like `modelId`, `datasets`, `co2_reported`, `createdat`, and `libraryname` were removed because of their lack of importance in the analysis. `database_efficiency` was also removed because it is a dependent variable of `co2_eq_emissions` and `size`. The first step was to choose metrics for classification, as the database *per se* does not have a relevant dependent variable to be classified. Therefore, some labels were created with the aim of creating a dependent variable to be predicted by a couple of machine learning models. Accordingly, `model_type` is the label used to determine which metric the model uses for its performance score, because not all models use the same metric for evaluation. After applying this classification, some rows were unusable because hundreds of them do not have performance metrics at all, so they were deleted. After the redundant rows were erased, a count was made on the qualitative variables, as shown in the following table:

TABLE I: model_type counts

model type	count
type1 (accuracy & f1)	773
type2 (rouge)	228
type3 (accuracy)	72
type4 (rouge1)	3
type5 (f1)	2

In an effort to find relationships between model types and the other categories, a summary of the other variables was prepared, focusing on the `domain` variable, which is the model's use case. The following table shows each domain and the associated percentage:

TABLE II: percentage of domains

model type	count
NLP	81,5%
Computer Vision	16,0%
Not specified	2,5%

In conclusion, there are only two major types of domain models. For the purpose of this analysis, this is not an influential variable, so it was removed. Other variables, like `library_name`, revealed that all models use `pytorch` as a library, so this was also removed. The same reasoning applied to `geographical_location`: the analysis revealed that 99% of the models do not specify the training location. Similarly, 99.7% of the models do not report the hardware used in `environment`. After applying analysis upon the mean of performance metrics and CO_2 emissions, the analysis revealed that those models could be labeled again with a more precise metric: *fairness*, defined as a relation between performance score and CO_2 emissions. The following are the labels chosen for the models

1) Fair and Efficient
2) Powerful but Expensive
3) Green but Weak
4) Inefficient

Finally, models were classified by a new boolean variable "is_fair", and those that were previously labeled with respective performance metrics, were evaluated: Results are consigned in the route: `/notebooks/part2/1-univarated-analysis.ipynb` Those classifications are required for the next step in the methodology, Cluster analysis.

B. Cluster Analysis

A grid over *distance* $\times$ *linkage* found that the highest internal separation was achieved by **Single–Euclidean** with $k = 3$ (silhouette ≈ 0.62, cophenetic ≈ 0.71). Close contenders were *Complete–Mahalanobis* ($k = 3$, silhouette ≈ 0.52, cophenetic ≈ 0.71) and *Average–Mahalanobis* ($k = 3$, silhouette ≈ 0.43, cophenetic ≈ 0.81). Among the "non-single" options, **Average–Euclidean** with $k = 4$ offered a more conservative structure (silhouette ≈ 0.34, cophenetic

TABLE III: Model Classification based on their fairness and performance metrics

Model Type	Classification Fairness	Count
type1 (accuracy & f1)	Fair and Efficient	229
	Powerful but Expensive	214
	Green but Weak	181
	Inefficient	142
	Incomplete data	7
type2 (rouge)	Inefficient	140
	Green but Weak	57
	Powerful but Expensive	18
	Fair and Efficient	9
	Incomplete data	4
type3 (accuracy)	Fair and Efficient	50
	Powerful but Expensive	9
	Green but Weak	7
	Incomplete data	4
	Inefficient	2
type4 (f1)	Incomplete data	2
type5 (rouge1)	Incomplete data	3

≈ 0.81). *Ward–Euclidean* showed the lowest separation (silhouette ≈ 0.20, cophenetic ≈ 0.44).

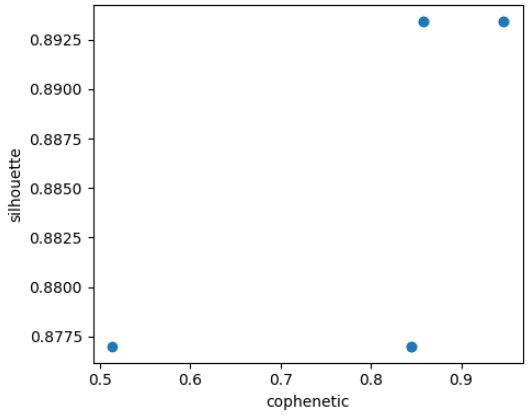

Fig. 1: Cophenetic vs. silhouette for all configurations. Top-right indicates better fidelity and separation.

Because *single* linkage may form chaining clusters, we report two complementary views: (i) the best-silhouette solution (**Single–Euclid,** $k = 3$) and (ii) a robust alternative without single linkage (**Average–Euclid,** $k = 4$). The horizontal cut at the recorded equivalent threshold reproduces the same k in the dendrogram.

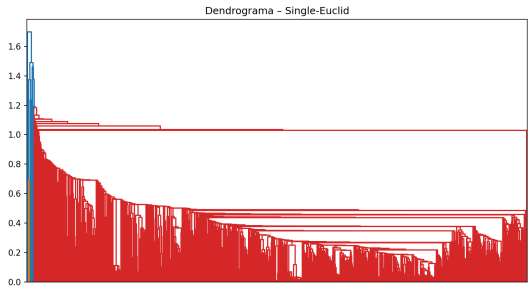

Fig. 2: Dendrogram—Single–Euclidean. The horizontal cut at the equivalent threshold yields $k = 3$.

Across configurations, a *dominant* "catalog" cluster concentrates the bulk of models near global means, while 1–3 *small* "elite" clusters group items with markedly higher *downloads/likes/performance* and (on average) lower *size/*CO_2. Variable importance from z–centroids is consistent: the most discriminative feature is **downloads** (max $|z| \approx 5.39$), followed by **likes** (≈ 2.03), then CO_2 and *performance* (moderate), with *size* contributing less (≈ 1.32).

External validation against labels was limited, as expected (labels were not used to train the clustering): ARI vs. `is_fair` stayed near zero across settings; however, cluster–label contingency showed reasonable purities (mean $\sim$ 0.73–0.90 depending on the configuration), with `is_fair=True` enriched in one of the small clusters and scarce in the large catalog group.

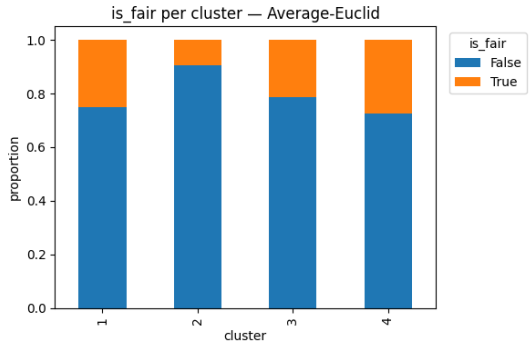

Fig. 3: Stacked proportions of `is_fair` per cluster (Average–Euclid, $k = 4$).

a) Cluster profiles.: Tables IV–VI report *original-unit* profiles (count/mean/median) for *performance_score*, *co2_eq_emissions*, *likes*, *downloads* and *size*. For brevity, we include the robust alternative (*Average–Euclid,* $k = 4$); the z–centroids and the $|z|$ ranking (downloads $\rightarrow$ likes) appear in the Supplement.

b) Overall quality.: Across linkages and distances, the highest internal separation was obtained by *Single–Euclid* with $k = 3$ (silhouette ≈ 0.62, cophenetic ≈ 0.71), followed by *Complete–Mahalanobis* (silhouette ≈ 0.52). Average-based variants trailed behind (silhouette $\approx$ 0.30–0.44) and *Ward–Euclid* showed the lowest separation (silhouette $\approx$ 0.20).

TABLE IV: Cluster profiles (original units): performance and CO_2

		performance_score		co2_eq_emissions	
cluster	**n**	**mean**	**median**	**mean**	**median**
1	4	0.988	0.991	49.974	3.189
2	1 029	0.727	0.822	71.382	2.749

TABLE V: Cluster profiles (original units): likes and downloads

		likes		downloads	
cluster	**n**	**mean**	**median**	**mean**	**median**
1	4	2.250	0.500	104 439.25	102 572.50
2	1 029	0.181	0.000	65.245	5.00

c) Cluster profiles (what differentiates groups).: Tables IV–V report original-unit profiles for a robust reference solution. Consistent with the winning configuration, a small cluster concentrates very high *downloads/likes* (and comparatively lower *size*/CO_2), while a large "catalog" cluster sits near the global mean. In z–space, *downloads* shows the largest between-cluster contrast ($\max|z| \approx 5.39$), followed by *likes* (≈ 2.03); CO_2, performance and size contribute more modestly (≈ 1.3–1.7).

d) External validation (not used for training).: Contingency tables indicate a small–to–moderate association between clusters and external labels. For *Single–Euclid* ($k = 3$), the largest cluster concentrates the majority of *is_fair=False* (share of *True* ≈ 0.27), whereas minor clusters show either higher *True* share or are too small to be conclusive. (mean/weighted), χ^2 with Cramér's V (≈ 0.03–0.07 across configs), and Fisher's exact test when applicable, together with ARI vs. *is_fair* (near zero, as expected for weak alignment). Figure 3 visualizes the *is_fair* proportions per cluster.

e) Sensitivity and limitations.: The pattern is robust across $k \in [2, 6]$ and across distance/linkage families that avoid overly compact (*ward*) or overly chained (*single*) artifacts; nonetheless, cluster sizes remain unbalanced and the *single* linkage can induce chaining in dense regions. These caveats do not affect the substantive finding that *downloads* (and, secondarily, *likes*) drive the segmentation.

C. Univariate Analysis

We ran k-means on the standardized variables (*performance_score*, *co2_eq_emissions*, *likes*, *downloads*, *size*) for $k \in \{2, \ldots, 10\}$ with `k-means++`, n_{init}=50, `max_iter`=500. Model selection combined four internal criteria:

- **Silhouette (max is best).** The curve peaked at $\mathbf{k = 2}$ with $\approx \mathbf{0.286}$, and decreased thereafter (Fig. 4).
- **Calinski–Harabasz (max is best).** The maximum occurred at k=3 (≈ 330), with k=2 close behind (≈ 290) (Fig. 4).
- **Davies–Bouldin (min is best).** DBI decreased monotonically with k, reaching ≈ 1.21 at k=10 (Fig. 5).

TABLE VI: Cluster profiles (original units): size ($\times 10^8$)

cluster	**n**	**mean**	**median**
1	4	2.649	2.654
2	1 029	8.953	4.987

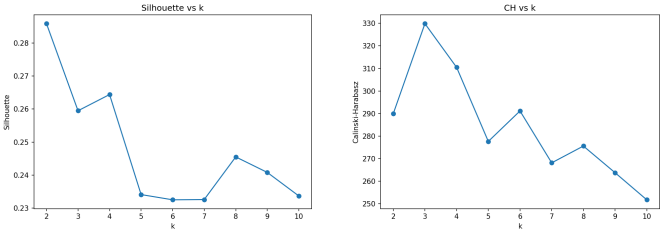

Fig. 4: Silhouette and Calinski–Harabasz across k.

- **Elbow (inertia).** Inertia dropped rapidly up to $k\sim6$ and then flattened, with no sharp elbow afterwards (Fig. 5).

Given the primary criterion (silhouette), we selected $\mathbf{k = 2}$. The resulting partition produced **unbalanced cluster sizes** (about $n\sim740$ vs. $n\sim300$; Fig. 6). Cluster profiles (in original units) and z–centroids were computed; the ranking by $\max|z|$ again highlighted *downloads* as the most discriminative variable, followed by *likes*, with the remaining variables contributing less. Cross-tabs against the external labels (*is_fair*, *clasification_fairness*, *model_type*) were generated for k=2 (see Supplement for the full tables).

The internal criteria point to different trade-offs: silhouette clearly favors k=2, Calinski–Harabasz prefers k=3, and Davies–Bouldin keeps improving as k grows. In this context, we prioritized silhouette because it balances compactness and separation and is less biased by the growth of k than CH/DBI. The *elbow* curve shows diminishing returns beyond $k\sim6$, supporting the choice of a small number of groups.

The k=2 solution is *interpretable* and *stable* (across restarts) but *unbalanced*: one major segment (catalog-like) and a smaller, high-intensity segment. The profile tables indicate that the minor cluster concentrates substantially higher *downloads* and *likes* (and slightly higher *performance*), while medians for *size*/CO_2 are not dominant—consistent with the hierarchical analysis. The $\max|z|$ ranking confirms *downloads* (then *likes*) as the strongest separators, which aligns with the business intuition that usage/engagement variables drive the segmentation more than footprint or size.

Regarding *external* labels, the cross-tabs for k=2 show heterogeneous mixes rather than perfectly pure clusters, i.e., they are useful for characterization but should not be read as supervised classes. This is expected because those labels were not used for training. Overall, the k-means segmentation complements the hierarchical results: it recovers the same two-segment story (catalog vs. high-traction niche), is easier to deploy, and preserves the same ordering of discriminative variables. If more granularity is needed, k=3 is a reasonable alternative per CH, though with a slight loss in silhouette and added complexity.

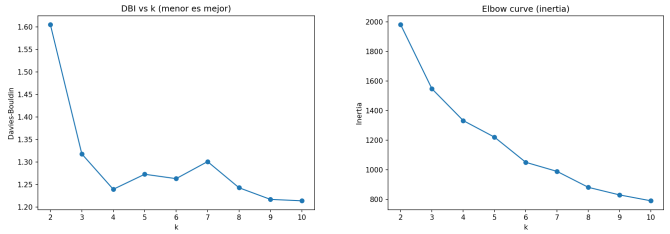

Fig. 5: Davies–Bouldin (lower is better) and inertia (elbow) across k.

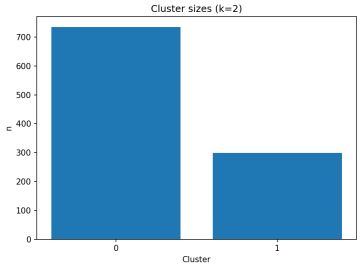

Fig. 6: Cluster sizes for the selected solution (k=2).

D. Multivariate Normality & LDA Feasibility

Visual diagnostics. Mahalanobis QQ–plots (global and by class) depart markedly from the χ^2_p reference line in the upper tail. The global plot shows a pronounced upward bend, indicating heavy–tailed behavior and/or mixture structure. Class–conditional plots (for `is_fair=False/True`) display the same pattern rather than aligning with the diagonal, so non–normality is not restricted to a single group.

Likely drivers. The heaviest deviations coincide with variables that are naturally skewed and highly dispersed in this domain (e.g., `downloads`, `likes`, and `size`). Even after z–scoring, extreme observations inflate the robust Mahalanobis distances, consistent with long right tails.

Implications for LDA. Classical LDA assumes (i) multivariate normality within each class and (ii) equal covariance matrices across classes. The QQ–plots provide clear evidence against (i), and the differing tail behavior between classes suggests that (ii) may also be questionable. Consequently, a standard LDA fit would risk biased boundaries and over–optimistic error estimates.

Recommended course. If a linear boundary is desired, prefer *regularized/shrinkage LDA* (e.g., Ledoit–Wolf within–class covariance) and consider mild preprocessing (log or $\log(1+x)$ transforms for strictly positive features; winsorization of top quantiles). As a distribution–free baseline, *penalized logistic regression* provides a linear separator without normality assumptions. If class scatters differ substantially, *QDA with regularization* or tree–based methods are more appropriate.

Summary. The MVN assumption is *not* supported for these features (globally nor within classes). Standard LDA is therefore not recommended without transformations and regularization; robust or nonparametric alternatives are better aligned with the observed data structure.

E. Análisis de covarianza y correlación

Para realizar el análisis de covarianza y correlación, lo primero es estandarizar los datos usando `StandardScaler` para presentarlos en la gráfica de manera más amigable, dado que las variables tienen una dispersión muy alta, a continuación se muestra la gráfica de correlación, ya que la de covarianza no permite ver las relaciones de manera tan clara

Para interpretar las correlaciones, se toma de manera univariada las relaciones entre las variables del conjunto de datos

1) `co2_eq_emissions` vs. `likes`: Con una correlación de 0.99, esta es una correlación positiva extremadamente fuerte. Sugiere que los modelos de IA que tienen un mayor número de likes (`likes`) también tienden a tener una emisión de CO2 equivalente mucho más alta.

2) `co2_eq_emissions` vs. `size`: Con una correlación de 0.99, similar al caso anterior, existe una correlación positiva extremadamente fuerte. Esto implica que los modelos de IA más grandes en términos de tamaño (`size`) son los que emiten una mayor cantidad de CO2.

3) `co2_eq_emissions` vs. `downloads`: Con una correlación de 0.04, La correlación es prácticamente cero (muy débil). Esto indica que no hay una relación lineal significativa entre el número de descargas (`downloads`) y la cantidad de emisiones de CO2. Un modelo puede tener muchas descargas independientemente de su huella de carbono.

4) `co2_eq_emissions` vs. `size_efficency`: Con una correlación de -0.01, La correlación es también cercana a cero. Esto sugiere que la eficiencia del tamaño (`size_efficency`) no está linealmente relacionada con las emisiones de CO2. Es posible que esta variable no esté capturando la eficiencia en términos de emisiones, o que la relación sea no lineal.

F. Análisis de componentes principales (PCA) y factores principales (PFA)

Para este apartado, se excluyeron las variables cualitativas, y se conservó de las ejecuciones anteriores la base de datos de imputados, limpios y estandarizados, luego de estandarizar los datos, se le aplicó la función `fit_transform` los detalles de la implementación se presentan en el notebook que se encuentra en la ruta: `3_analisis_factores.ipynb` del proyecto.

1) Análisis de componentes principales: El primer resultado que se encuentra es acerca de la varianza explicada, los resultados se encuentran consolidados en la siguiente tabla:

TABLE VII: Resultados del Análisis de Componentes Principales

Componente	Valor propio	Varianza Explicada	Varianza Acumulada
PC1	2.989	59.74%	59.74%
PC2	1.012	20.23%	79.97%
PC3	0.988	19.75%	99.72%
PC4	0.010	0.21%	99.93%
PC5	0.003	0.07%	100.00%

2) Interpretación de los resultados:

- **Componente 1**: Con un valor propio de `2.989`, este componente es el más importante. Explica casi el 60%

de la varianza total de los datos por sí solo. Esto significa que la mayor parte de la información está concentrada en esta primera dimensión.
- **Componente 2**: Explica un `20.23%` de la varianza. Al combinarlo con el Componente 1, ambos componentes juntos explican casi el `80%` de la varianza acumulada.
- **Componente 3**: Este componente explica casi un 20% de la varianza. Al incluirlo, la varianza acumulada asciende al `99.72%`.
- **Componentes 4 y 5**: Estos componentes tienen valores propios muy pequeños y explican una varianza insignificante (menos del 1% entre ambos).

3) Selección de componentes: Para decidir cuántos componentes conservar, se aplicó el criterio de Kaiser, que sugiere mantener los componentes con un valor propio mayor a 1. Según este criterio, se deberías considerar conservar solo los dos primeros componentes principales, ya que sus valores propios son `2.989` y `1.012`. Estos dos componentes juntos resumen el `79.97%` de la varianza de tus datos, conservarlos permite reducir la dimensionalidad del conjunto de datos de cinco variables a solo dos, perdiendo muy poca información esencial. A continuación, se presenta la matriz de cargas factoriales para los componentes principales. Cada valor indica la correlación de la variable original con el componente correspondiente.

TABLE VIII: Resultados del Análisis de Componentes Principales

Variable Original	Carga en PC1	Carga en PC2
`co2_eq_emissions`	0.577	0.047
`downloads`	0.033	-0.695
`likes`	-0.031	0.715
`size`	-0.126	0.052
`size_efficency`	0.805	0.031

4) Interpretación de los Componentes Principales Clave:
- **Componente 1 (PC1)**: Este componente, que explica la mayor parte de la varianza del conjunto de datos, está fuertemente asociado con las variables 4 y 0. Ambas contribuyen de manera positiva a este componente, sugiriendo que el PC1 captura un factor común de estas dos variables.
- **Componente 2 (PC2)**: Este componente representa un contraste entre la variable 2 y la variable 1. La alta carga positiva de la variable 2 y la alta carga negativa de la variable 1 indican que el PC2 describe una relación inversamente proporcional entre ellas."
- **Varianza en el PC1**: La mayoría de los puntos se agrupan cerca de la coordenada 0 en el eje horizontal, con la notable excepción de un punto que se encuentra muy alejado, alrededor de PC1 = 60. Este punto atípico (o outlier) es una observación con un valor extremadamente alto en el primer componente principal. La gran dispersión a lo largo del eje horizontal refuerza lo que vimos en el análisis de los valores propios: el PC1 captura la mayor parte de la varianza en tus datos.
- **Varianza en el PC2**: En el eje vertical (PC2), los puntos también están bastante concentrados cerca de la coordenada 0, aunque hay una mayor dispersión vertical en este grupo principal. Hay un par de puntos que se alejan del grupo, pero la variabilidad es considerablemente menor en comparación con el PC1. Esto confirma que el PC2 explica menos varianza que el PC1.
- **Presencia de un Outlier**: La observación aislada en la esquina inferior derecha es un caso extremo. Este punto tiene un valor muy alto en el PC1 y un valor moderadamente bajo en el PC2. Es probable que esta observación represente un modelo de IA con características extremas en las variables originales que más contribuyen al PC1 (por ejemplo, un tamaño o una emisión de CO2 excepcionalmente grandes).

5) Análsis de factores principales (PFA): Para el Análisis Factorial de Principales Componentes (PFA), los números que ha obtenido son los valores propios (eigenvalues) de los factores. A diferencia del PCA, donde los valores propios explican la varianza total de los datos, en el PFA estos valores explican solo la varianza común (o compartida) entre las variables, lo cual es la varianza que puede atribuirse a factores subyacentes. La interpretación de los resultados se presenta a continuación en la siguiente tabla:

TABLE IX: Interpretación de Factores Principales

Factor	Valor propio	% de Varianza Común Explicada	% Acumulado
`F1`	2.987	74.49%	74.49%
`F2`	1.012	25.26%	99.75%
`F3`	0.988	0.22%	99.97%
`F4`	0.010	0.03%	100.00%
`F5`	0.003	0.00%	100.00%

6) Interpretación de factores (PFA):
- **Factor 1 (Valor Propio: 2.987)**: Este factor captura la mayor parte de la varianza compartida, explicando un notable 74.49% de ella por sí solo. Es, con diferencia, el factor más importante.
- **Factor 2 (Valor Propio: 1.012)**: Este factor también es muy significativo, ya que explica otro 25.26% de la varianza común. Al combinar el Factor 1 y el Factor 2, se explica casi el 99.75% de toda la variabilidad compartida de tu conjunto de datos.
- **Factores Restantes (3, 4 y 5)**: Los valores propios de estos factores son muy bajos, lo que indica que explican una cantidad insignificante de la varianza común

Para determinar cuantos factores retener, se usa el criterio de kaiser, el detalle de la implementación puede verse en la ruta: `notebooks/3_analisis_factores.ipynb`

7) Análisis de cargas factoriales: El Factor 1 es el más significativo, tal como lo vimos en el análisis de los valores propios. Las variables con las cargas más altas en este factor son:

- `co2_eq_emissions`: `0.999200`
- `likes`: `0.995705`
- `size`: `0.993753`

Estos valores son extremadamente altos, lo que significa que el Factor 1 es, en esencia, un resumen de estas tres variables. Todas ellas tienen una correlación positiva muy

fuerte con el factor, lo que sugiere que los modelos con un gran tamaño y un alto número de likes tienden a tener una alta emisión de CO2. Este factor puede ser interpretado como un factor de "Complejidad y Popularidad del Modelo" que está directamente relacionado con la huella de carbono. Las variables `downloads` y `size_efficency` tienen cargas muy bajas en el Factor 1, lo que indica que no contribuyen significativamente a este factor latente. El Factor 2 explica la varianza común restante. La variable con la carga más alta en este factor es `downloads`: `0.718192`. Esto sugiere que el Factor 2 está definido principalmente por el número de descargas. Las otras variables tienen cargas muy bajas, lo que indica que no están fuertemente correlacionadas con este factor. Por lo tanto, el Factor 2 puede ser interpretado como un factor de "Alcance o Adopción del Modelo", que es conceptualmente distinto del primer factor. Los resultados confirman que las variables de emisiones de CO2, likes y tamaño están altamente correlacionadas y se agrupan en el mismo constructo subyacente, mientras que las descargas representan un concepto separado el conjunto datos.

El gráfico de sedimento confirma visualmente lo que los valores numéricos indicaban: los Factores 1 y 2 son los más significativos y los únicos que vale la pena conservar para el análisis. Los factores 3, 4 y 5 tienen valores propios muy bajos, lo que significa que explican una cantidad trivial de varianza y pueden ser descartados. A continuación se presentan los biplots de los métodos PCA y PFA

8) Interpretación de Biplots: **Para el biplot de factores principales**: **Factor1 popularidad y tamaño:** Este factor, representado por el eje horizontal, está fuertemente correlacionado con las variables likes, `co2_emissions` y `size`. Las aplicaciones con valores altos en este factor tienden a ser más populares, con una mayor cantidad de 'me gusta', un tamaño de archivo más grande y, posiblemente, un mayor impacto ambiental. **Factor 2** Capacidad de Descarga: este factor, representado por el eje vertical, se relaciona principalmente con la variable downloads. Indica que las aplicaciones con altas puntuaciones en este factor son las que tienen un alto número de descargas, independientemente de su popularidad o tamaño. La mayoría de las observaciones se agrupan en el centro, lo que sugiere que la mayoría de las aplicaciones tienen características promedio. Sin embargo, se pueden identificar dos grupos distintos de aplicaciones:

- Las que se caracterizan por una alta popularidad y tamaño (Factor 1).
- Las que se distinguen por un alto número de descargas (Factor 2).

La variable size_efficiency muestra una correlación muy baja con ambos factores, lo que indica que no es un buen predictor de la popularidad, el tamaño o las descargas de una aplicación. En cuanto al biplot de análisis de componentes principales. El PC1, que explica la mayor parte de la varianza en los datos, establece un contraste directo entre likes y downloads. Esto indica una fuerte correlación negativa: las aplicaciones con un alto número de "me gusta" tienden a tener pocas descargas, mientras que aquellas con muchas descargas reciben menos "me gusta". Esto podría sugerir dos estrategias distintas de éxito. Además, el PC2 no es un factor dominante, ya que las variables size, size_efficiency y co2_eq_emissions tienen una baja correlación con él. Esto significa que estas variables no aportan valor para explicar la principal variabilidad del conjunto de datos. y En síntesis: El biplot revela que la varianza de los datos se explica principalmente por una sola dimensión: el equilibrio entre la popularidad social y el volumen de descargas. Las otras variables tienen un impacto mínimo en la forma en que tus datos se agrupan.

IV. Conclusión

El análisis comparativo de las técnicas multivariadas aplicadas al conjunto de datos, específicamente el Análisis de Componentes Principales (PCA) y el Análisis de Factores Principales (PFA), ha revelado información crucial sobre la estructura subyacente de las variables.

A. Justificación del Método Seleccionado

El Análisis de Factores Principales (PFA) se identifica como el método más apropiado para este estudio. Aunque el PCA logró identificar una dimensión de varianza predominante, su principal objetivo es la reducción de dimensionalidad y no la interpretación de constructos latentes. En contraste, el PFA, diseñado para tal fin, proporcionó un modelo más robusto y conceptualmente interpretable.

B. Hallazgos encontrados

El modelo de PFA extrajo satisfactoriamente dos factores latentes que explican las interrelaciones observadas entre las variables. Estos factores se interpretan de la siguiente manera:

Factor 1: Popularidad y Escala. Este factor se correlaciona fuertemente con las variables likes, co2_emissions y size. Dicho factor representa un constructo de éxito que agrupa atributos de popularidad y tamaño.

Factor 2: Capacidad de Adquisición. Este factor está casi exclusivamente definido por la variable downloads. Su independencia del Factor 1 sugiere que la capacidad de una aplicación para ser descargada es una dimensión distinta y no redundante del éxito.

C. Implicaciones

Los resultados demuestran que el comportamiento de las aplicaciones no se puede explicar adecuadamente por una sola dimensión, sino que está influenciado por al menos dos constructos subyacentes. La identificación de estos factores permite una comprensión más profunda de la dinámica del mercado y proporciona una base sólida para la toma de decisiones estratégicas, orientadas a optimizar el rendimiento en cada una de estas dimensiones independientes.

D. Conclusión final

Aunque el conjunto de datos es funcional para un análisis exploratorio, su limitada granularidad y la ausencia de estandarización en las métricas de evaluación impiden un análisis riguroso de la huella de carbono. La variabilidad en el hardware utilizado, la ubicación geográfica de los centros

de datos y la falta de consistencia en los informes de consumo energético y emisiones de CO2 representan obstáculos significativos para identificar con certeza las variables más influyentes. Estos factores comprometen la capacidad de extraer conclusiones definitivas sobre la relación entre las configuraciones de los modelos de inteligencia artificial y su impacto ambiental. Por lo tanto, se enfatiza la necesidad de futuros estudios que se beneficien de conjuntos de datos más detallados, actualizados y estandarizados. Un enfoque más riguroso en la recopilación de datos permitirá una comprensión más profunda del impacto ambiental de la IA y facilitará el desarrollo de prácticas más sostenibles en la industria.

REFERENCES

[1] UST, "What is green ai?" https://www.ust.com/en/ust-explainers/what-is-green-ai, 2023, accessed: 11 de septiembre de 2025.

[2] ——, "Exploring the carbon footprint of hugging face's ml models: A repository mining study," https://arxiv.org/pdf/2305.11164, 2023, accessed: 11 de septiembre de 2025.