**Trabajo Práctico Individual Minería de Datos**

**Sistemas Inteligentes**

A colorful crest with a bird and a dolphin

AI-generated content may be incorrect.

Gallo Kuhn Mauro Andrés.

2025.

Universidad del Salvador

Facultad de Ingeniería

Contenido

[Objetivo 3](#_Toc201860852)

[Dataset 3](#_Toc201860853)

[Link 3](#_Toc201860854)

[Descripción 3](#_Toc201860855)

[Cantidad de Ejemplos y de Atributos 4](#_Toc201860856)

[Tareas 4](#_Toc201860857)

[Limpieza 4](#_Toc201860858)

[Selección 7](#_Toc201860859)

[Integración 8](#_Toc201860860)

[Transformación 8](#_Toc201860861)

[Minería 10](#_Toc201860862)

[Evaluación 11](#_Toc201860863)

[Conclusiones 16](#_Toc201860864)

# Objetivo

El objetivo de este trabajo práctico es aplicar el proceso de Descubrimiento de Conocimiento en Datos (KDD) sobre un dataset real, utilizando técnicas de clasificación supervisada. El propósito principal es predecir la personalidad de una persona (introvertida o extrovertida) en base a sus conductas sociales, aplicando distintas etapas del proceso KDD como limpieza, transformación, minería y evaluación de modelos.

Para ello se utilizó Python junto a la biblioteca scikit-learn, trabajando en un entorno Jupyter Notebook que permitió visualizar cada paso del proceso y tomar decisiones en función de los datos y resultados obtenidos.

# Dataset

## Link

[Extrovert vs. Introvert Behavior Data - Kaggle](https://www.kaggle.com/datasets/rakeshkapilavai/extrovert-vs-introvert-behavior-data)

## Descripción

El dataset utilizado recopila información sobre conductas sociales y hábitos cotidianos vinculados a la personalidad. Está orientado a identificar patrones de comportamiento asociados a los rasgos de extroversión e introversión. Las variables reflejan factores como el tiempo que una persona pasa sola, la frecuencia con la que asiste a eventos sociales o publica en redes, entre otros.

A screenshot of a black screen

AI-generated content may be incorrect.

*Vistazo general del dataset*

Cada fila representa a una persona, y el atributo clasificador (*Personality*) clasifica si su perfil es predominantemente extrovertido o introvertido.

## Cantidad de Ejemplos y de Atributos

* Total de ejemplos: 2.900
* Total de atributos: 8
  + 7 Atributos predictivos.
  + 1 Atributo clasificador.

# Tareas

## Limpieza

Se comenzó con una inspección inicial del dataset para visualizar la cantidad de ejemplos disponibles, sus atributos y el tipo de dato que manejaban estos atributos.

A computer screen shot of a black screen

AI-generated content may be incorrect.

Al igual que se revisó qué tan balanceado estaba el dataset entre ambas clases, para así determinar más adelante si era necesario pasar por algún algoritmos que balanceara mejor el dataset y decidir la estrategia de entrenamiento y validación.

A graph with blue and red bars

AI-generated content may be incorrect.

Los resultados reflejaron que el dataset se encontraba bastante balanceado (51.4% Extrovertidos, 48.6% Introvertidos), por lo que no iba a ser necesario aplicar alguna técnica específica, e incluso una división tradicional iba a ser suficiente.

Durante esta inspección también se revisaron aquellos atributos duplicados, atributos con valores nulos, y la cantidad de ejemplos totales que tenían al menos uno de sus atributos con valores nulos.

A partir de ahí, se obtuvo la siguiente información:

A computer screen shot of a black screen

AI-generated content may be incorrect.

Por un lado, se detectaron varias filas duplicadas dentro del dataset. Sin embargo, dado que los datos fueron recolectados mediante un formulario con opciones limitadas, es esperable que distintas personas hayan proporcionado respuestas idénticas. Estas observaciones duplicadas representan respuestas reales y válidas, por lo que no fueron eliminadas. Por este motivo, se optó por conservarlas para mantener la integridad del dataset.

Por otro lado, el resto de información ya nos estaba indicando que habían datos nulos los cuales necesitaban ser limpiados. Existían dos opciones:

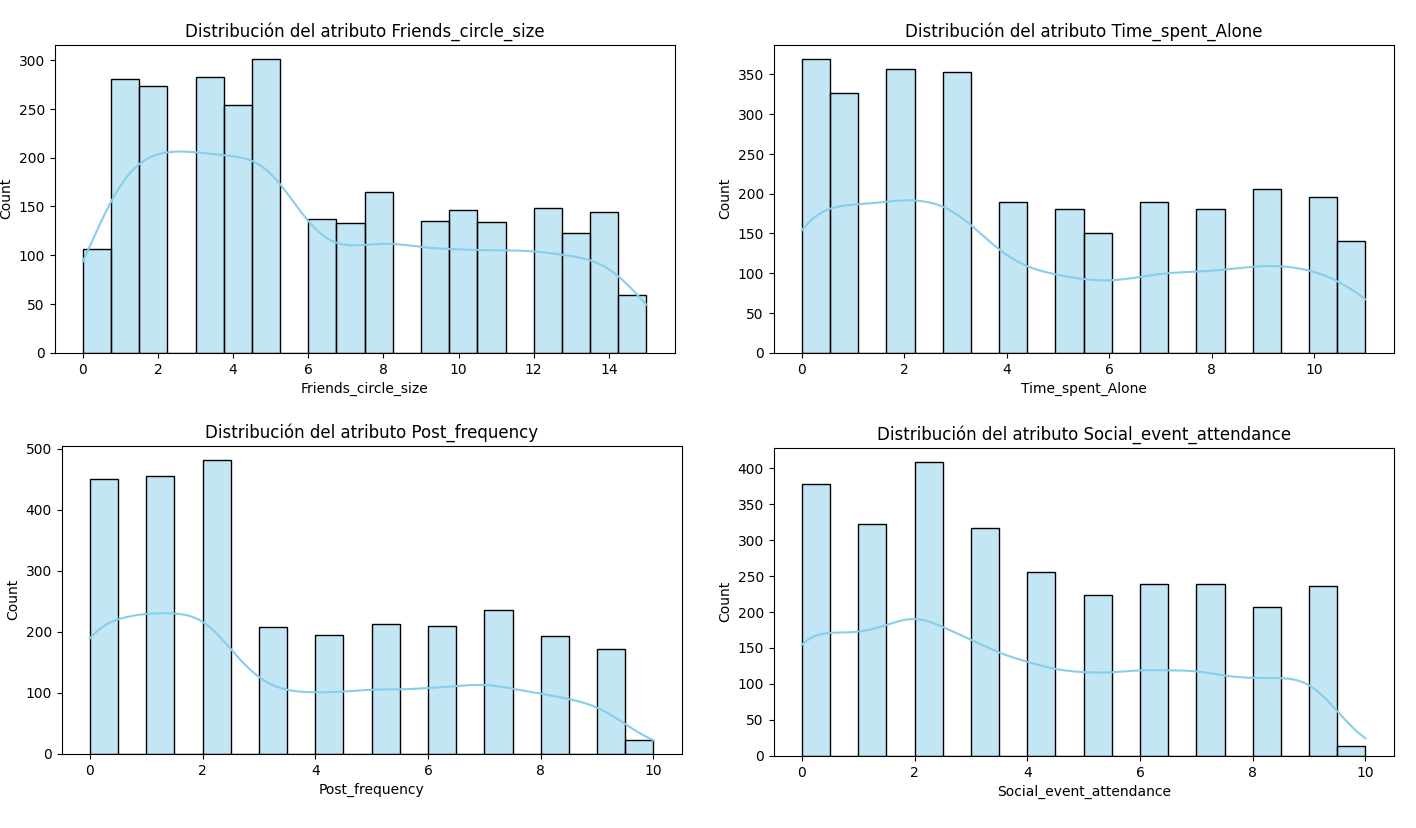
* Eliminar todos aquellos ejemplos con datos nulos.
* Imputar los valores faltantes.

La primera opción parecía la más directa, pero eliminar 423 ejemplos implicaría quitar aproximadamente un 14,5% del total del dataset, lo cual iba a quitar información sumamente valiosa para el modelo.

Por ende, se optó por imputar los valores faltantes, tanto para los atributos continuos como para los atributos categóricos.

*Imputación de atributos continuos*

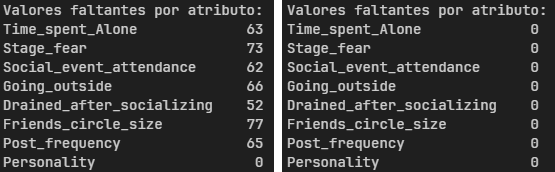
A la hora de imputar los atributos continuos, se podía utilizar tanto la media como la mediana, siendo la segunda un parámetro más robusto dado que contempla valores atípicos y con una distribución asimétrica, las cuales son casi siempre el fenómeno más típico en datasets de este tipo. De todas formas, se procedió a analizar la distribución de los atributos y se confirmó que todas las distribuciones eran asimétricas.



*Ejemplo de distribución para cuatro atributos continuos del dataset*

*Imputación de atributos categóricos*

Para imputar los atributos categóricos, se utilizó la moda para calcular el valor más frecuente en cada atributo.



*Antes y después del proceso de limpieza*

De esta forma, se cubrieron todos los valores nulos del dataset, y estaba todo listo para pasar a la fase de selección.

## Selección

Se decidió utilizar el archivo personality\_dataset.csv (y no el personality\_datasert.csv), ya que el primero presentaba valores nulos, lo que permitía aplicar técnicas de limpieza y preprocesamiento como parte del aprendizaje.

Los atributos seleccionados fueron los 7 que no corresponden al clasificador, dado que se consideró que todos influían de manera importante en la determinación del tipo de personalidad de un individuo:

* Time\_spent\_Alone
* Stage\_fear
* Social\_event\_attendance
* Going\_outside
* Drained\_after\_socializing
* Friends\_circle\_size
* Post\_frequency

Y como atributo clasificador, se eligió:

* Personality (Introvert / Extrovert)

## Integración

No fue necesario integrar fuentes de datos externas, ya que todo el análisis se realizó sobre un único archivo .csv consistente. No se requirió combinación con otras tablas ni enriquecimiento externo.

## Transformación

Se realizaron las siguientes transformaciones sobre el dataset para prepararlo correctamente antes del modelado:

*Codificación binaria*

Los atributos categóricos con valores binarios como Stage\_fear y Drained\_after\_socializing fueron convertidos de valores tipo texto (Yes/No) a valores numéricos binarios (1/0). De esta forma los modelos de machine learning podrán interpretar los datos correctamente.

El atributo clasificador Personality, también fue transformado y se codificó como 1 para Extrovert y 0 para Introvert.

A screenshot of a computer

AI-generated content may be incorrect.

*Escalado de atributos*

Antes de avanzar con el modelado, se consideró la necesidad de escalar los atributos numéricos del dataset. Este paso resulta clave para que todas las variables estén en una misma escala y así evitar que aquellas con valores numéricamente más grandes influyan de manera desproporcionada en ciertos algoritmos sensibles a la magnitud de los datos (como K-Nearest Neighbors o Regresión Logística). De este modo, se asegura que todas las variables aporten de forma equitativa en el cálculo de distancias o en los procesos internos de cada modelo.

En el caso de los atributos binarios, el escalado no era estrictamente necesario porque ya estaban normalizados entre 0 y 1. Sin embargo, se decidió aplicar el procedimiento a todo el dataset por consistencia, sin que esto genere efectos negativos sobre el rendimiento.

Se evaluaron distintas alternativas:

* MinMaxScaler: transforma los valores dentro de un rango fijo (usualmente 0–1), aunque es sensible a valores extremos.
* StandardScaler: ajusta cada atributo para que tenga media 0 y desviación estándar 1, lo que lo hace más robusto frente a valores atípicos y más adecuado cuando las variables no tienen un rango máximo definido.
* Normalizer: escala fila por fila y se emplea en casos donde importa el ángulo entre vectores (texto, series temporales).

Se optó por StandardScaler porque:

* El dataset incluye variables continuas no acotadas a un rango específico (ejemplo: tamaño de amistades de 0 a 15, tiempo en soledad de 0 a 11).
* Se observó cierta asimetría en la distribución de algunos atributos.
* Los algoritmos seleccionados son sensibles a la escala y este método ofrece un balance entre estabilidad y resistencia a valores extremos.

Es importante remarcar que el escalado no se aplicó de forma global al dataset completo, sino dentro de cada etapa de minería, específicamente en los respectivos training sets tras realizar los splits. Esto evita la filtración de información del conjunto de prueba en el proceso de normalización y garantiza que las métricas obtenidas no estén sesgadas artificialmente.

A modo ilustrativo, así se visualizarían los atributos continuos una vez escalados:



## Minería

En esta etapa se aplicaron distintos algoritmos de clasificación con el objetivo de predecir el tipo de personalidad (Introvertido o Extrovertido).

Como ya habíamos determinado que el dataset se encontraba muy bien balanceado, se optó por realizar una división simple en el dataset de 80% para entrenamiento y 20% para test, procurando que en cada subset (training set y test set) haya la misma proporción de extrovertidos e introvertidos.

Luego de realizar las divisiones, se entrenaron los siguientes modelos:

*Regresión Logística*

Fue el primer modelo probado por su simplicidad, rapidez y buen rendimiento en problemas con relaciones lineales entre los atributos y la variable a predecir.

*Random Forest*

Este modelo construye varios árboles de decisión, cada uno entrenado con diferentes partes del dataset, y luego promedia sus predicciones para obtener un resultado más robusto. Funciona bien en casos donde puede haber relaciones no lineales entre los atributos.

*K-Nearest Neighbors*

Es un modelo basado en distancias. Clasifica un ejemplo según la mayoría de clases presentes en los K vecinos más cercanos. En este caso se usó K=5, un valor común para comenzar.

## Evaluación

Para evaluar el rendimiento de los modelos entrenados, se usaron las métricas más comunes en tareas de clasificación binaria:

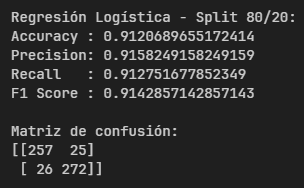
* Accuracy.
* Precision.
* Recall.
* F1 Score.

Se aplicaron tres enfoques de evaluación:

*División 80-20*

Para la primera iteración, se entrenó cada modelo con el 80% del dataset y se evaluó con el 20% restante. Esto ayudó a obtener una primera idea de rendimiento general. Los resultados para cada modelo fueron:

* Regresión Logística: Mostró un buen rendimiento para todas las métricas.



* Random Forest: Mostró un rendimiento levemente inferior a Regresión Logística, pero con buenos resultados.

A screenshot of a computer

AI-generated content may be incorrect.

* K-Nearest Neighbors: Logró un rendimiento muy similar a Regresión Logística, levemente superior en precisión, pero levemente inferior en Recall.

A screenshot of a computer

AI-generated content may be incorrect.

*Cross-Validation*

Para la segunda iteración, y asegurar una mayor robustez en los resultados, se aplicó la técnica de cross-validation con 5 particiones, obteniendo las métricas promedio de los 5 entrenamientos y tests.

Este método ayudó a reducir el sesgo que podía llegar a ocurrir por una única división de datos, y brindó un resultado más confiable del desempeño de los modelos. En líneas general, los 3 modelos demostraron una mejora muy leve, reflejando que el dataset se encontró bien estructurado desde un inicio.

* Regresión Logística:

A screenshot of a computer

AI-generated content may be incorrect.

* Random Forest:

A screenshot of a computer

AI-generated content may be incorrect.

* K-Nearest Neighbors:

A screenshot of a computer

AI-generated content may be incorrect.

*Cross-Validation + Grid Search*

Para la última iteración, se buscó aumentar todavía más las métricas de los modelos, en particular la métrica Accuracy dado que, por el contexto del problema y el buen balance del dataset, es la que mejor refleja los resultados obtenidos.

Para eso, se utilizó la técnica de Grid Search, donde se busca la mejor combinación de parámetros para cada modelo individual, con el objetivo de refinarlos aún más y tener resultados mejores. Se configuró el Grid Search para que evalúe múltiples combinaciones de hiperparámetros, considerando el rendimiento en varias métricas para evitar configuraciones desbalanceadas. Finalmente, se seleccionó aquella con el mayor Accuracy.

Luego se volvió a ejecutar el modelo utilizando cross-validation con los hiperparámetros optimizados.

* Regresión Logística:

A screenshot of a computer

AI-generated content may be incorrect.

* Random Forest:

A screenshot of a computer program

AI-generated content may be incorrect.

* K-Nearest Neighbors:

A screenshot of a computer

AI-generated content may be incorrect.

En general, se puede visualizar cómo las métricas de los tres modelos aumentaron, reflejando que el uso de esta técnica es muy recomendado.

# Conclusiones

Este trabajo permitió aplicar en profundidad el proceso KDD sobre un caso práctico de clasificación binaria. Se comenzó desde el análisis del dataset y la limpieza de datos, hasta llegar al modelado, evaluación y validación cruzada de los resultados obtenidos.

Uno de los principales desafíos fue decidir cómo tratar los valores nulos. En lugar de eliminar datos, se optó por imputar valores para no perder información, utilizando técnicas adecuadas para datos numéricos y categóricos. Esto contribuyó a conservar la integridad del dataset y mejorar el entrenamiento de los modelos.

El proceso de transformación también fue clave, ya que requirió convertir variables categóricas en valores numéricos, y luego escalar todos los atributos para que los modelos pudieran interpretarlos correctamente.

Durante la fase de minería, se probaron tres modelos distintos: Regresión Logística, Random Forest y K-Nearest Neighbors. Esta diversidad permitió comparar enfoques tanto lineales como no lineales, y entender mejor cómo cada modelo responde a los mismos datos.

Por último, se aplicaron tres formas de evaluación: una división tradicional 80-20 y otra mediante cross-validation, lo cual permitió tener una visión más certera del desempeño de cada modelo.

En general, se logró cumplir el objetivo del trabajo: predecir la personalidad de los individuos en función de sus comportamientos sociales, evaluando los modelos aplicados y comprendiendo el impacto de cada paso del proceso. La experiencia también sirvió para afianzar conceptos prácticos del ciclo de minería de datos y de machine learning aplicado a datos reales.