Laboratorio 8
Exploración y Uso Avanzado de Plataformas IA, Repositorios Profesionales y Herramientas Globales para el desarrollo de IA y de SW

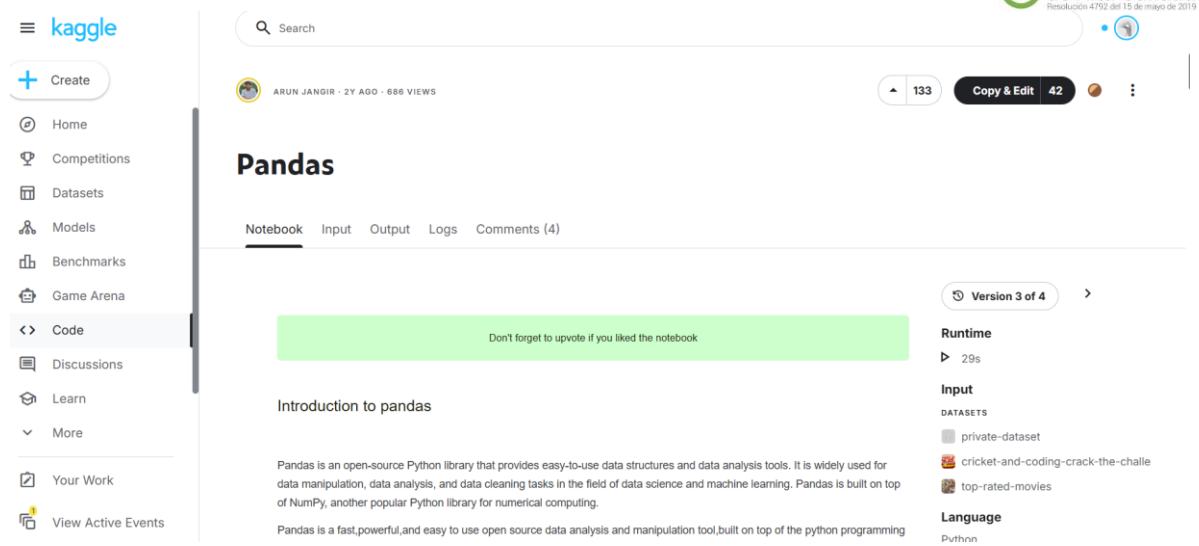
Materia:
Profundización de inteligencia artificial

Participantes:
Ana Maria Navarro

Profesor:
Carlos Betancourt correa

Universidad de Manizales
ingeniería en sistemas y telecomunicaciones
Manizales, Caldas, Colombia

2.4 Kaggle – Datasets, experimentos y colaboración



1. Introducción

Kaggle es una plataforma líder en ciencia de datos que ofrece datasets públicos, notebooks ejecutables en la nube y un entorno colaborativo para aprendizaje y experimentación con machine learning. En el contexto del Laboratorio 8, el objetivo de esta actividad es realizar un Análisis Exploratorio de Datos (EDA) sobre un dataset, aplicar técnicas básicas de limpieza y visualización, y publicar el notebook como evidencia en la plataforma, documentando luego los resultados y sus posibles implicaciones.
2025_1_IA_1_Lab_08_Repositorios…

2. Descripción del dataset

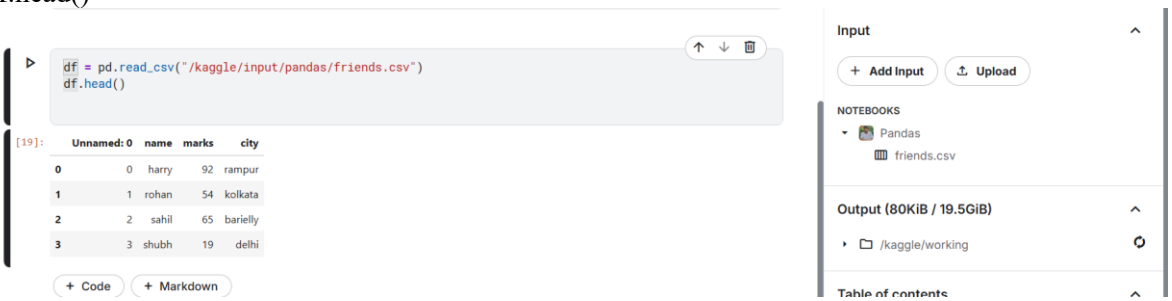
Para esta actividad se utilizó un dataset sencillo llamado friends.csv, disponible dentro del notebook de Kaggle. El dataset contiene información de 4 estudiantes, con las siguientes columnas principales:

- name: nombre del estudiante.
- marks: calificación numérica obtenida.
- city: ciudad de residencia.
- Unnamed: 0: índice generado automáticamente (columna irrelevante para el análisis).

Este tipo de datos, aunque pequeño, es representativo de un escenario educativo donde se registran las notas de estudiantes y su procedencia.

3. Carga del dataset

En el notebook se importó la librería pandas y se leyó el archivo CSV:
df = pd.read_csv("/kaggle/input/pandas/friends.csv")
df.head()



La instrucción df.head() permitió visualizar las primeras filas, comprobando que los datos se cargaron correctamente y que las columnas estaban bien definidas.

4.Exploración inicial

Se obtuvo información general del dataset:
df.info()

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4 entries, 0 to 3
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Unnamed: 0  4 non-null      int64 
 1   name        4 non-null      object
 2   marks       4 non-null      int64 
 3   city        4 non-null      object
dtypes: int64(2), object(2)
memory usage: 260.0+ bytes
```

+ Code + Markdown

Lo que mostró:

- 4 registros (filas).
- 4 columnas.
- Ningún valor nulo en el dataset.
- Tipos de datos:
  - Unnamed: 0 y marks: int64
  - name y city: object

Esto confirmó que los datos eran pequeños, completos y adecuados para un EDA introductorio.

5. Estadísticos descriptivos

Se calcularon estadísticas con:

df.describe(include="all")

```
df.describe(include="all")
```

```
/usr/local/lib/python3.11/dist-packages/pandas/io/formats/format.py:1458: RuntimeWarning: invalid value encountered in greater
  has_large_values = (abs_vals > 1e6).any()
/usr/local/lib/python3.11/dist-packages/pandas/io/formats/format.py:1459: RuntimeWarning: invalid value encountered in less
  has_small_values = ((abs_vals < 10 ** (-self.digits)) & (abs_vals > 0)).any()
/usr/local/lib/python3.11/dist-packages/pandas/io/formats/format.py:1459: RuntimeWarning: invalid value encountered in greater
  has_small_values = ((abs_vals < 10 ** (-self.digits)) & (abs_vals > 0)).any()
```

[21]:

	Unnamed: 0	name	marks	city
count	4.000000	4	4.00000	4
unique	NaN	4	NaN	4
top	NaN	harry	NaN	rampur
freq	NaN	1	NaN	1
mean	1.500000	NaN	57.50000	NaN
std	1.290994	NaN	30.22692	NaN
min	0.000000	NaN	19.00000	NaN
25%	0.750000	NaN	45.25000	NaN
50%	1.500000	NaN	59.50000	NaN
75%	2.250000	NaN	71.75000	NaN
max	3.000000	NaN	92.00000	NaN

+ Code + Markdown

A pesar de aparecer algunos warnings de formato, el resultado permitió obtener:

- Para marks:
  - **Media** alrededor de 57.5
  - **Mínimo** próximo a 19
  - **Máximo** cercano a 92
- Para las variables categóricas (name, city):
  - Conteo de registros.
  - Valores únicos presentes.

Estas estadísticas permitieron tener una primera idea de la distribución de notas: se observan estudiantes con calificaciones altas y otras significativamente más bajas.

6. Limpieza de datos

Aunque el dataset no tenía valores nulos (df.isnull().sum() devolvió 0 en todas las columnas), sí se identificó una columna innecesaria:

- **Unnamed: 0** → índice redundante.

```
df.isnull().sum()
```

```
[22]: Unnamed: 0    0
      name          0
      marks         0
      city          0
      dtype: int64
```

+ Code + Markdown

Se procedió a eliminarla con:
df = df.drop(columns=["Unnamed: 0"])
df.head()

```
df = df.drop(columns=["Unnamed: 0"])
df.head()
```

[23]:

	name	marks	city
0	harry	92	rampur
1	rohan	54	kolkata
2	sahil	65	barielly
3	shubh	19	delhi

+ Code + Markdown

Con esto se obtuvo un dataframe más limpio, con solo las columnas relevantes: name, marks, city.

7. Visualizaciones

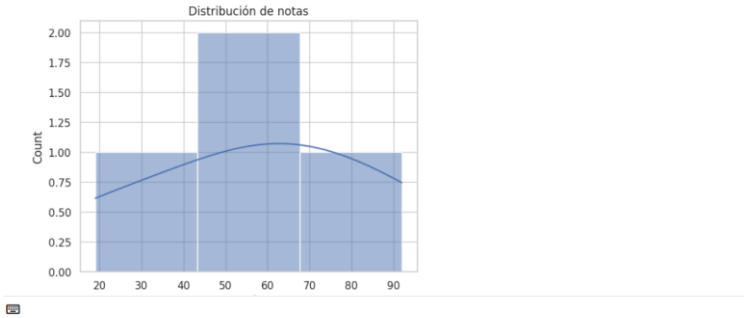
Para el análisis exploratorio se crearon varias gráficas utilizando **seaborn** y **matplotlib**:

7.1 Distribución de notas

sns.histplot(df["marks"], kde=True)
plt.title("Distribución de notas")
plt.show()

```
sns.histplot(df["marks"], kde=True)
plt.title("Distribución de notas")
plt.show()
```

```
/usr/local/lib/python3.11/dist-packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



La gráfica de histograma mostró cómo se distribuyen las calificaciones de los cuatro estudiantes. Se observa que:

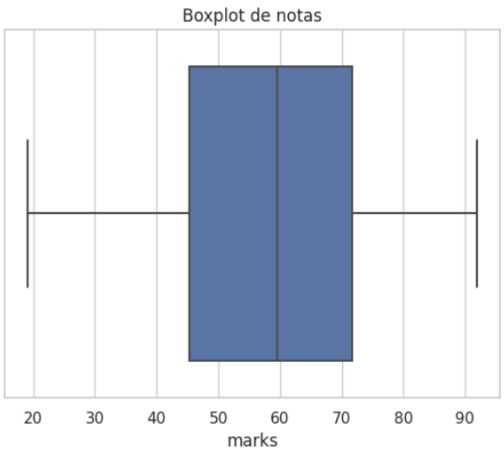
- Hay una nota muy alta (cercana a 90).
- Hay una nota muy baja (alrededor de 20).
- Las otras dos se encuentran en un rango medio.

Esto sugiere una **alta variabilidad** entre el desempeño de los estudiantes.

7.2 Boxplot de notas

sns.boxplot(x=df["marks"])
plt.title("Boxplot de notas")
plt.show()

```
sns.boxplot(x=df['marks'])
plt.title("Boxplot de notas")
plt.show()
```



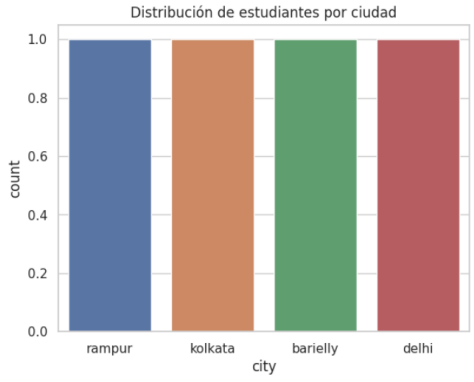
El boxplot permitió identificar valores extremos y la dispersión general. Se aprecia:

- Una mediana en un rango intermedio.
- Un rango amplio entre mínimo y máximo, consistente con el histograma.
- Comportamiento que podría considerarse heterogéneo para un grupo tan pequeño.

7.3 Distribución de estudiantes por ciudad

sns.countplot(x=df["city"])
plt.title("Distribución de estudiantes por ciudad")
plt.show()

```
sns.countplot(x=df['city'])
plt.title("Distribución de estudiantes por ciudad")
plt.show()
```

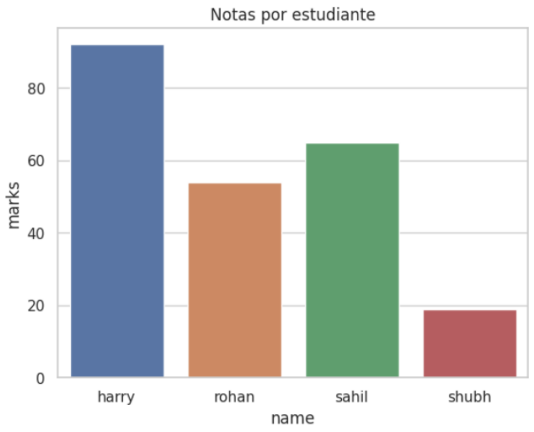


En este caso, cada ciudad (rampur, kolkata, barielly, delhi) aparece exactamente una vez, lo que indica que el dataset está balanceado por ciudad (cada estudiante proviene de una ciudad diferente).

7.4 Notas por estudiante

sns.barplot(x=df["name"], y=df["marks"])
plt.title("Notas por estudiante")
plt.show()

```
sns.barplot(x=df['name'], y=df['marks'])
plt.title("Notas por estudiante")
plt.show()
```



Esta gráfica de barras permitió visualizar de manera clara qué estudiante obtuvo la mayor nota y quién obtuvo la menor:

- Un estudiante destaca notablemente con la calificación más alta.
- Otro presenta un desempeño muy bajo.
- Los restantes se encuentran en un rango medio.

8. Conclusiones

- Se realizó un EDA completo sobre un dataset sencillo de calificaciones de estudiantes.
- Se verificó la ausencia de valores nulos y se eliminó una columna irrelevante, dejando los datos limpios.
- Se calcularon estadísticas descriptivas que permitieron entender la distribución de las notas.
- Se construyeron visualizaciones (histograma, boxplot, countplot, barplot) que facilitaron la interpretación de los resultados.
- El notebook fue publicado en Kaggle, cumpliendo con los requerimientos del laboratorio respecto a documentación y colaboración en la plataforma.