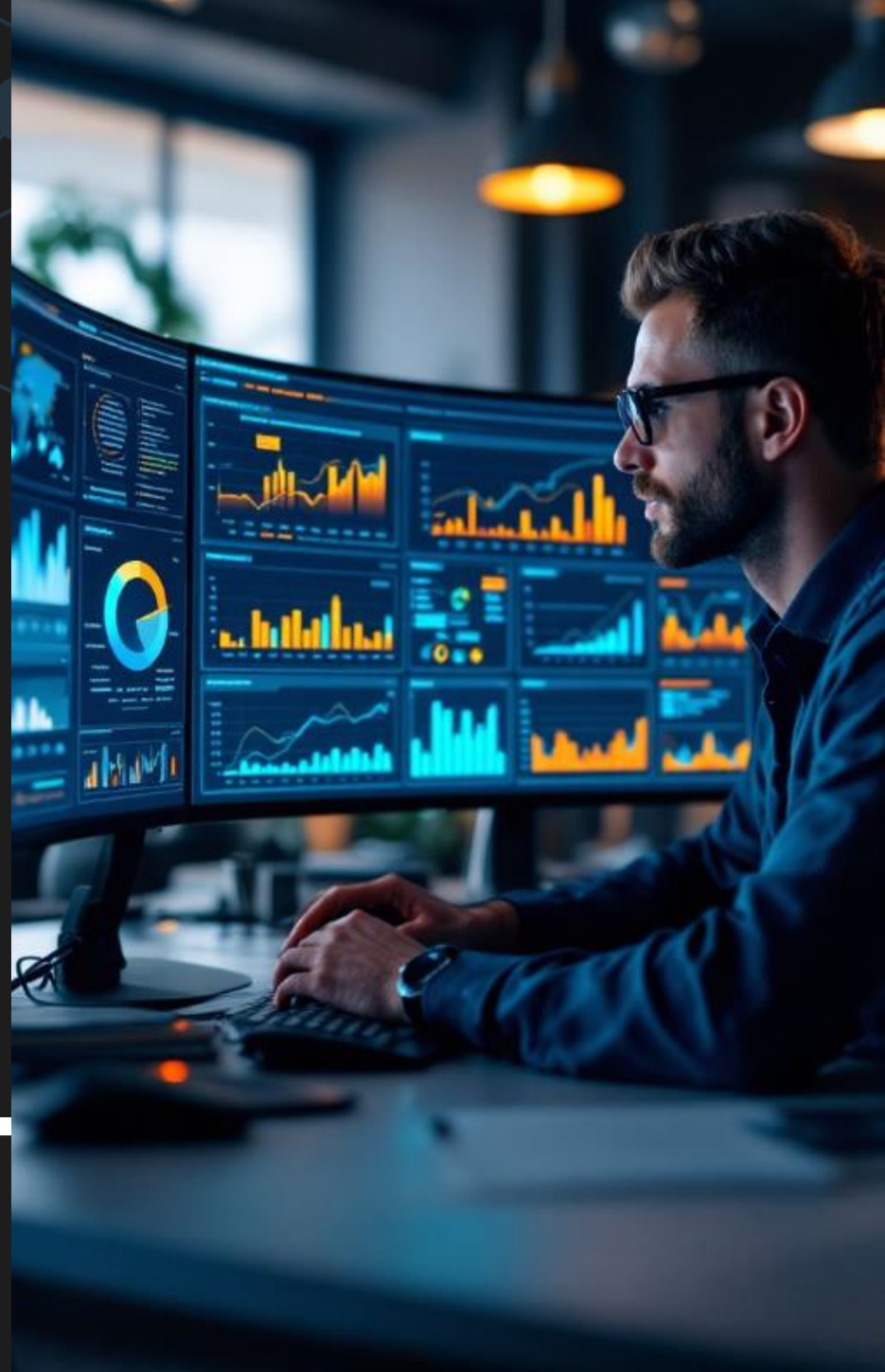
Análisis Exploratorio

de Datos

Sesión 1

Análisis Exploratorio de Datos (EDA)

- El Análisis Exploratorio de Datos es una etapa fundamental en cualquier proyecto de ciencia de datos. Consiste en la exploración inicial de los datos para comprender su estructura, identificar patrones, detectar anomalías y formular hipótesis que guiarán decisiones sobre técnicas de análisis o modelado a aplicar.
- Este proceso no solo nos ayuda a familiarizarnos con los datos, sino que también es crucial para garantizar la calidad de nuestros análisis posteriores y la validez de nuestros modelos predictivos.



Objetivos del EDA

1

Comprender la estructura y distribución

Permite visualizar cómo están organizados los datos, su tipo, forma, y comportamiento general. Esto incluye identificar la distribución de variables y detectar tendencias iniciales o agrupaciones.

2

Detectar valores atípicos y datos faltantes

Ayuda a identificar valores extremos o inconsistencias, así como registros incompletos que podrían distorsionar los análisis y deben ser tratados de forma adecuada.

3

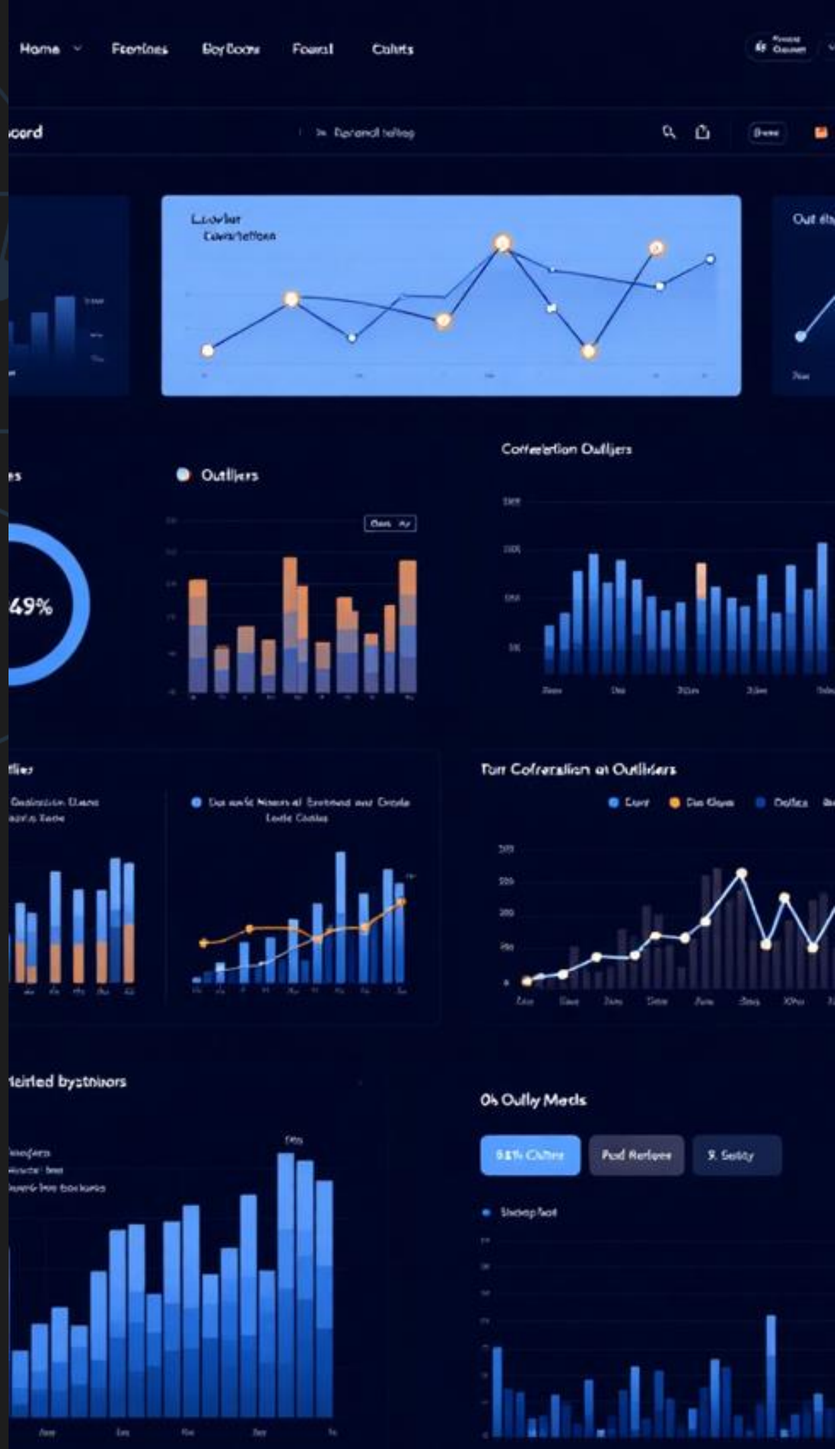
Descubrir relaciones entre variables

Permite analizar correlaciones, patrones conjuntos o dependencias entre columnas que puedan tener valor explicativo o predictivo.

4

Formular hipótesis

Facilita la generación de preguntas e ideas sobre los datos que orienten futuros análisis, modelos estadísticos o decisiones basadas en evidencia.



Análisis Inicial de Datos (IDA)

Carga de datos
Importar el conjunto de datos desde archivos CSV, Excel, SQL u otras fuentes. Este paso establece la base para todo el análisis posterior y requiere verificar que los datos se carguen correctamente.

Inspección inicial
Revisar las primeras filas, tipos de datos y dimensiones del conjunto de datos. Esto proporciona una visión general de la estructura y contenido de los datos disponibles.

Identificación de problemas
Detectar valores faltantes, duplicados o inconsistencias en los datos que podrían afectar la calidad del análisis posterior y requerir limpieza o transformación.

El Análisis Inicial de Datos (IDA) es una fase preliminar del EDA que se enfoca en la inspección básica de los datos para asegurar que estén listos para análisis más profundos.

Ejemplo de IDA con Pandas

```python
import pandas as pd

# 1 Cargar el conjunto de datos desde un archivo CSV
df = pd.read_csv('datos_ejemplo.csv')  # Reemplaza con la ruta de tu archivo

# 2 Inspección inicial: Ver las primeras filas, tipos de datos y dimensiones
print("Primeras filas del dataset:")
print(df.head(), "\n")  # Muestra las primeras 5 filas

print("Información del dataset:")
print(df.info(), "\n")  # Muestra los tipos de datos y las dimensiones del dataset

# 3 Identificación de problemas:
# a) Valores faltantes
print("Valores faltantes por columna:")
print(df.isnull().sum(), "\n")  # Muestra el número de valores faltantes por columna

# b) Duplicados
print("Duplicados en el dataset:")
print(df.duplicated().sum(), "\n")  # Muestra la cantidad de filas duplicadas

# c) Estadísticas descriptivas para detectar valores atípicos
print("Estadísticas descriptivas:")
print(df.describe(), "\n")  # Muestra estadísticas como media, desviación estándar, etc.

# d) Tipos de datos para revisar posibles conversiones
print("Tipos de datos:")
print(df.dtypes, "\n")  # Muestra los tipos de datos de cada columna
```

Carga de datos: Usamos pd.read_csv() para cargar los datos desde un archivo CSV. Asegúrate de que el archivo esté en la misma carpeta que el script o ajusta la ruta del archivo.

Inspección inicial:

- df.head() nos muestra las primeras 5 filas para entender la estructura del conjunto de datos.
- df.info() nos da información sobre el número de filas, columnas y el tipo de datos de cada columna.

Identificación de problemas:

- Valores faltantes: df.isnull().sum() nos ayuda a ver si alguna columna tiene datos faltantes.
- Duplicados: df.duplicated().sum() nos dice si hay filas duplicadas que deben ser eliminadas.
- Estadísticas descriptivas: df.describe() nos proporciona una vista general de las estadísticas que nos puede ayudar a detectar valores atípicos.
- Tipos de datos: df.dtypes nos muestra los tipos de datos de cada columna.

Técnicas Claves del EDA

Análisis Univariado

Estudia una sola variable a la vez para entender su distribución y comportamiento. Utiliza histogramas, gráficos de caja (boxplots) y medidas estadísticas como media, mediana y moda para caracterizar cada variable individualmente.

Análisis Bivariado

Analiza dos variables juntas para detectar relaciones o patrones. Emplea diagramas de dispersión (scatter plots), correlación de Pearson y análisis de tablas cruzadas (crosstabs) para identificar dependencias entre pares de variables.

Análisis Multivariado

Estudia la relación entre múltiples variables simultáneamente. Utiliza matrices de correlación, gráficos de pares (pair plots) y técnicas como la regresión múltiple para descubrir interacciones complejas entre variables.

Visualización de Datos

Las herramientas de visualización ayudan a interpretar los datos de manera más efectiva. Incluye histogramas, diagramas de dispersión, gráficos de barras y mapas de calor (heatmaps) para representar visualmente patrones y tendencias.

Aplicaciones del EDA

Ciencia de Datos

Preparación de datos antes de entrenar modelos de machine learning, detección de valores atípicos y datos faltantes, y selección de características relevantes para modelado predictivo.

Negocios y Marketing

Identificación de tendencias en ventas y comportamiento del cliente, segmentación de clientes basada en análisis de patrones y detección de oportunidades de mercado a partir de datos históricos.

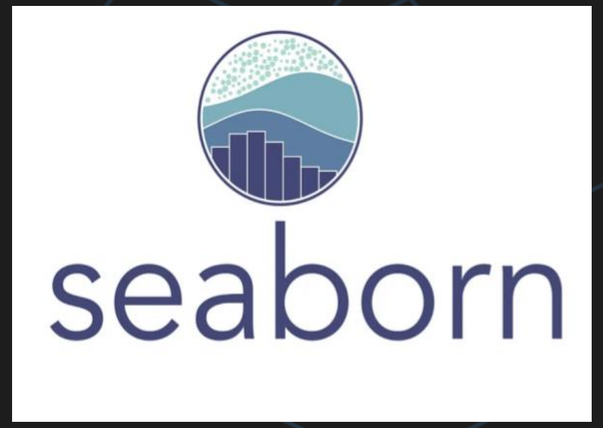
Finanzas y Contabilidad

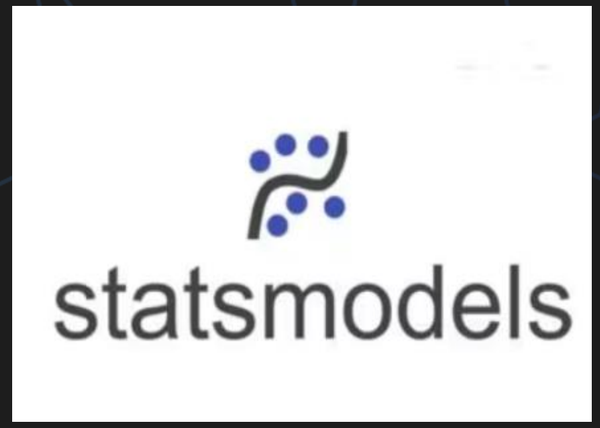
Detección de fraudes y anomalías en transacciones bancarias, análisis de riesgos financieros y evaluación de portafolios de inversión, y seguimiento de tendencias económicas.

Investigación Científica

Exploración de patrones en datos experimentales, comparación de distribuciones en estudios clínicos o sociales y validación de hipótesis antes de aplicar modelos estadísticos.

Herramientas para EDA





Las herramientas más populares para realizar Análisis Exploratorio de Datos incluyen:

- **Pandas** para manipulación de datos tabulares
- **NumPy** para cálculos numéricos
- **Matplotlib** y **Seaborn** para visualizaciones estadísticas
- **SciPy** y **Statsmodels** para análisis estadístico avanzado

Estas bibliotecas de Python se complementan entre sí para proporcionar un ecosistema completo que facilita la exploración, transformación y visualización de datos de manera eficiente y efectiva.

Herramientas para EDA

Statsmodels permite aplicar modelos estadísticos como regresiones lineales, ANOVA y pruebas de hipótesis. Además, proporciona herramientas para realizar **regresiones diagnósticas**, análisis de residuos, intervalos de confianza y test de significancia.

📌 **Ejemplo básico:**

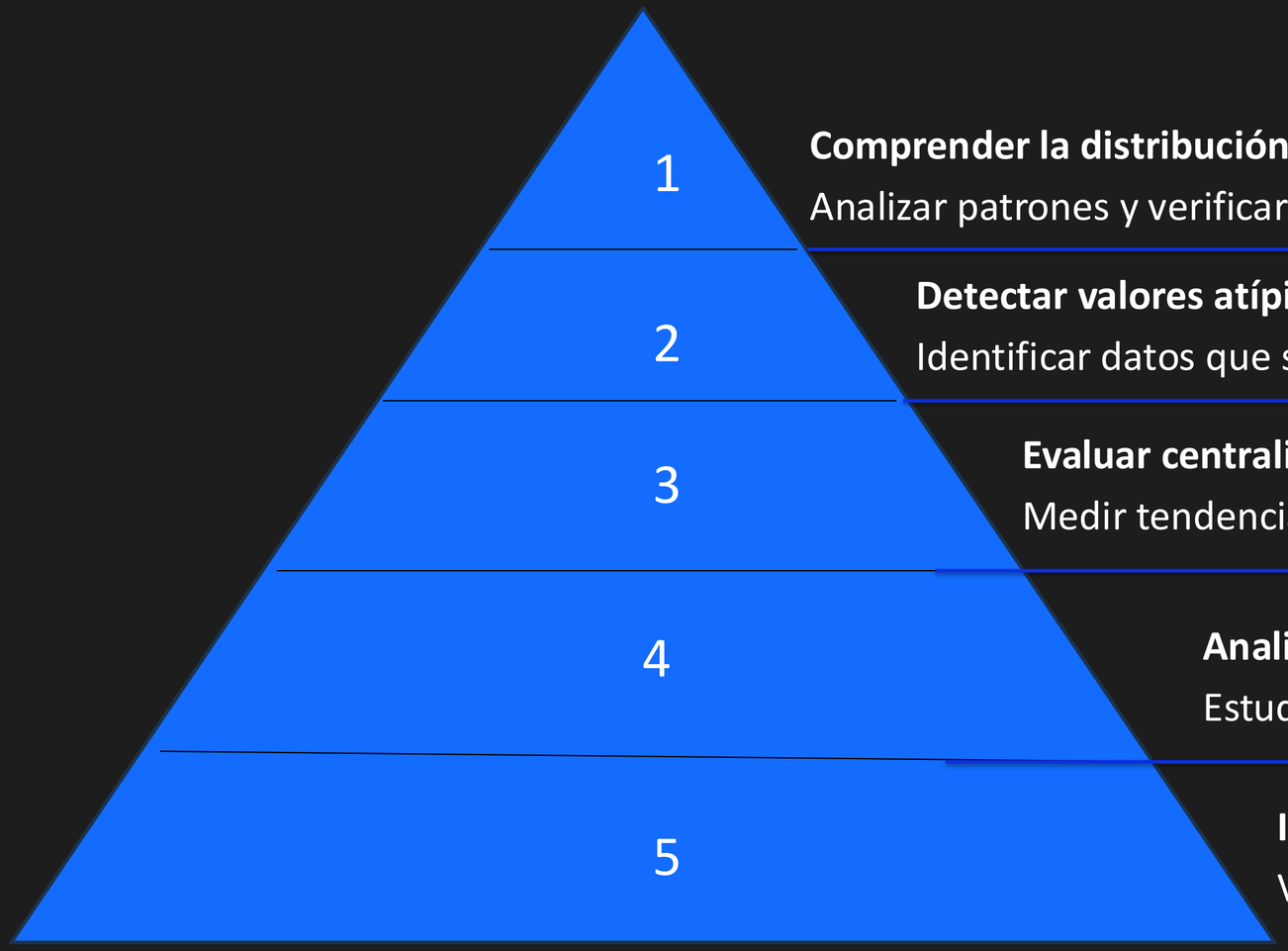
```
import statsmodels.api as sm

# Supongamos que queremos ajustar una regresión lineal simple
X = df['horas_estudio']
y = df['nota']

X = sm.add_constant(X)  # Agrega intercepto
modelo = sm.OLS(y, X).fit()

# Ver resumen estadístico del modelo
print(modelo.summary())
```

Análisis Univariado



Comprender la distribución
Analizar patrones y verificar distribuciones

Detectar valores atípicos
Identificar datos que se desvían significativamente

Evaluar centralidad
Medir tendencia central con media, mediana y moda

Analizar dispersión
Estudiar variabilidad con desviación estándar y rango

Identificar sesgos
Verificar distribuciones sesgadas que afecten interpretación

El Análisis Univariado examina una sola variable a la vez para comprender su comportamiento, distribución y características principales. Es fundamental en el EDA ya que proporciona una visión clara de cada variable antes de proceder a análisis más complejos.

Ejemplo de Análisis Univariado con Pandas

```python
# Calcular medidas estadísticas
print("Media de Precios:", df["Precio"].mean())
print("Mediana de Precios:", df["Precio"].median())
print("Desviación Estándar de Precios:", df["Precio"].std())

# Visualización con boxplot
sns.boxplot(x=df["Precio"])
plt.title("Boxplot de Precios")
plt.show()
```

Medidas de Tendencia Central y Dispersión

Calcula métricas clave como media, mediana, varianza y desviación estándar.

A continuación, se hace una visualización con boxplot.

¿Por Qué Usar Pandas para el Análisis Univariado?

✓Eficiencia: Herramientas rápidas y fáciles de implementar.

✓Flexibilidad: Adaptable a variables numéricas y categóricas.

✓Visualización Integrada: Combinación perfecta con Matplotlib y Seaborn para gráficos claros y profesionales.

Análisis Multivariado

✅ El Análisis Multivariado examina múltiples variables simultáneamente para identificar patrones, correlaciones y relaciones entre ellas. Es útil para evaluar cómo una variable depende de otra y detectar tendencias en conjuntos de datos complejos.

✅ Las técnicas más utilizadas incluyen la Correlación para evaluar relaciones entre variables numéricas, el Análisis de Componentes Principales (PCA) para reducción de dimensionalidad, y los Gráficos de Dispersión Matricial (Pair Plots) para visualizar relaciones entre pares de variables en una sola figura.

Técnicas de Análisis Multivariado

📌Correlación

- Evalúa la relación entre dos variables numéricas.
- Se usa el coeficiente de correlación de Pearson.

📌Análisis de Componentes Principales (PCA)

- Técnica de reducción de dimensionalidad para datos con muchas variables.

📌Gráficos de Dispersión Matricial (Pair Plots)

- Visualización de relaciones entre pares de variables en una sola figura.

Ejemplo de Análisis Multivariado

```python
# Matriz de correlación
print(df.corr())

# Visualización de un mapa de calor para correlaciones
plt.figure(figsize=(8,6))
sns.heatmap(df.corr(), annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Matriz de Correlación")
plt.show()

# Pair plot para analizar relaciones entre variables
sns.pairplot(df, hue="Categoría")
plt.show()
```

Matriz de Correlación

Para identificar las relaciones lineales entre variables, se calcula la matriz de correlación utilizando: *print(df.corr())*

Visualización con Mapa de Calor

Se utiliza un mapa de calor para representar gráficamente las correlaciones, facilitando la interpretación visual de las relaciones más fuertes.

Análisis de Pares de Variables con Pair Plot

Finalmente, se emplea un pair plot para explorar visualmente las relaciones bivariadas y distribuciones individuales de las variables.

¿Por Qué Usar estas herramientas?

✓ Matriz de Correlación: Proporciona una vista numérica clara de las relaciones entre variables.

✓ Mapa de Calor: Ofrece una representación visual intuitiva que resalta patrones y fuerza de correlaciones.

✓ Pair Plot: Ideal para analizar interacciones entre múltiples variables y detectar tendencias o agrupaciones.

Preguntas

Sección de preguntas

Análisis Exploratorio

de Datos

Continúe con las actividades