

## Prueba 2

### Instrucciones detalladas:

#### 1. Selección de la API pública y definición del problema

Cada estudiante o grupo deberá elegir una API pública relevante para el desarrollo de un caso de estudio con alto valor analítico.

- **Alpha Vantage:** Análisis de precios de acciones para predicción de tendencias. [Alpha Vantage](#)
- **OMDb API:** Análisis de películas para clasificación de géneros según características. [OMDb API](#)
- **WHO COVID-19 Data:** Predicción de la propagación de enfermedades. [Disease](#)

### Tareas:

- Definir un problema de negocio claro con base en los datos seleccionados
- Documentar los desafíos de los datos y sus implicaciones.

#### 2. Extracción y preprocesamiento de datos avanzados

Los estudiantes deberán:

1. Conectar a la API seleccionada utilizando [requests](#).
2. Almacenar datos en una base de datos temporal utilizando [SQLite](#) o en archivos CSV para análisis posteriores.
3. Aplicar técnicas de limpieza avanzadas, como manejo de valores nulos, transformación de formatos y eliminación de duplicados.
4. Realizar **ingeniería de características**, generando nuevas variables a partir de las existentes para enriquecer el análisis.

#### 3. Análisis exploratorio de datos (EDA) con visualización avanzada

Los estudiantes deben explorar los datos en profundidad utilizando las siguientes técnicas:

- Análisis de tendencias temporales mediante [rolling averages](#) y descomposición estacional de series temporales.
- Detección de **outliers** utilizando métodos como IQR o Z-Score.
- Análisis de correlación avanzada utilizando mapas de calor ([seaborn](#)).
- Gráficos avanzados como [pairplots](#), [time series decomposition](#), y distribuciones con [seaborn](#) y [matplotlib](#).

#### 4. Construcción de modelos predictivos avanzados

Los estudiantes deben:

1. Implementar modelos de **regresión** (si la variable objetivo es continua) o **clasificación** (si es categórica).
2. Aplicar **validación cruzada** para evaluar los modelos.

3. Probar múltiples algoritmos como Árboles de Decisión, Random Forest, XGBoost.
4. Evaluar el rendimiento con métricas como **MAE**, **RMSE**, **precisión**, **recall**, **F1-score**.

## 5. Generación de reglas de decisión con árboles de decisión

A partir del modelo generado, los estudiantes deberán interpretar las reglas extraídas del árbol de decisión para crear un sistema de reglas que permita la toma de decisiones.

### Ejemplo de regla generada:

*"Si la temperatura es mayor a 30°C y la humedad es inferior a 40%, entonces se recomienda encender el sistema de riego automático."*

### Tareas:

- Extraer las reglas del modelo con `sklearn.tree.export_text()`.
- Generar un sistema automatizado basado en las reglas para tomar decisiones.

## 6. Implementación de visualización interactiva

Los estudiantes deberán construir un **dashboard interactivo** utilizando **Dash** o **Streamlit** que incluya:

1. Predicción basada en entradas del usuario.
2. Visualización de tendencias en gráficos dinámicos.
3. Reglas de negocio aplicadas a los resultados obtenidos.

### Rúbrica.

Criterios	Peso	100%	80%	60%	30%	0%
<b>Extracción y limpieza de datos (API)</b>	20%	Se conecta exitosamente a la API, extrae y estructura los datos correctamente. Se aplican técnicas de limpieza avanzadas eliminando valores nulos y errores.	Se conecta a la API y extrae datos correctamente, con mínima limpieza de datos faltantes o errores.	La conexión es parcial o presenta errores, limpieza incompleta.	Conexión fallida y limpieza inadecuada.	No se realizó la extracción ni limpieza de datos.
<b>Análisis exploratorio de datos (EDA)</b>	20%	Exploración completa con visualizaciones relevantes, estadísticas descriptivas detalladas y correlaciones	Exploración aceptable con visualizaciones adecuadas pero sin análisis profundo de correlaciones.	Exploración básica con visualizaciones limitadas y poco análisis estadístico.	Exploración superficial sin uso adecuado de herramientas de visualización.	No se realizó el análisis exploratorio.

		analizadas correctamente .				
<b>Implementación de modelos predictivos y descriptivos</b>	30%	Se implementan correctamente modelos predictivos y descriptivos con evaluaciones precisas de desempeño (precisión, recall, F1-score).	Implementación adecuada, pero con errores menores en métricas de evaluación o selección de características .	Modelos con problemas de implementación y bajo rendimiento.	Modelos incompletos o sin evaluación adecuada.	No se implementaron modelos.
<b>Interpretación de resultados y sistema de reglas</b>	15%	Se identifican reglas claras y relevantes basadas en el análisis del modelo, con recomendaciones prácticas bien fundamentadas.	Se generan reglas adecuadas, pero con falta de profundidad en la interpretación.	Las reglas son generales y poco útiles para la toma de decisiones.	Reglas confusas sin relación clara con los resultados obtenidos.	No se generaron reglas.
<b>Presentación y visualización de resultados</b>	15%	Dashboard interactivo bien estructurado con visualizaciones claras y explicativas, reflejando correctamente los hallazgos.	Dashboard funcional pero con visualizaciones limitadas y falta de interactividad.	Dashboard básico sin interactividad ni claridad en la presentación.	Dashboard mal estructurado y sin relación con los hallazgos.	No se realizó ninguna visualización.

## Formato de entrega:

- Informe técnico en Nootbook** con:
  - Definición del problema.
  - Proceso de extracción y preprocesamiento.
  - Análisis exploratorio y hallazgos.
  - Modelos implementados y evaluación.
  - Conclusiones y recomendaciones.
- Notebook Jupyter** con código funcional y documentado.
- Presentación de resultados** en Notebook con hallazgos clave.