

Guía de Diseño de Analíticas y Modelos

Guía para el desarrollo y despliegue de analíticas y modelos en Aitea Building Lab



Índice de Contenidos

- 1 Introducción
- 2 ¿Qué es una analítica?
- 3 Metodología paso a paso
- 4 Conceptos técnicos: Transformaciones vs Modelos
- 5 Codificación de analíticas
- 6 Configuración y entrenamiento de modelos
- 7 Pruebas y versionado
8. Conclusiones

Introducción

Esta formación está diseñada para proporcionar una visión global sobre el diseño, implementación y validación de analíticas y modelos de IA en el contexto de edificios inteligentes usando Aitea Building Lab.

El desarrollo de analíticas robustas permite:

- ◆ Extraer información valiosa de los datos generados por sensores Automatizar
 - ◆ procesos de decisión basados en datos
- Optimizar el funcionamiento de edificios inteligentes

¿Qué es una Analítica?

Una **analítica** es un algoritmo, modelo o técnica de machine learning (incluidos cálculos estadísticos, probabilistas, algoritmos específicos del ML, incluidos modelos de deep learning) cuyo fin es extraer una información no evidente en los datos recogidos de los sensores que arroja un edificio inteligente.

Objetivo principal: Construir eventos informativos y/o desencadenar acciones automatizadas a partir del análisis de datos.

Características clave:

- Procesa datos provenientes de sensores en edificios inteligentes
- Utiliza técnicas de machine learning para identificar patrones no evidentes
- Transforma datos brutos en información accionable
- Se implementa mediante código en Aitea Building Lab
- Se integra en un sistema más amplio para la gestión de edificios (Aitea Brain Lite)

Metodología Paso a Paso

El desarrollo de analíticas en Aitea Building Lab sigue un proceso estructurado para garantizar soluciones efectivas:

- 1 Estudio del sistema y de los datos**
Análisis del sistema y los datos generados. Boceto de posibles aproximaciones indicando objetivos claros y finalidad de resultados. Incluye diseño de limpieza de datos.
- 2 Establecimiento de la analítica**
Selección de la aproximación más útil. Definición de datos de entrada/salida y determinación de conjuntos para entrenamiento y pruebas.
- 3 Documentación técnica**
Descripción formal de la analítica, incluyendo interfaz de entrada/salida y descripción detallada del algoritmo.
- 4 Codificación, pruebas y documentación**
Implementación, validación y documentación usando el framework Aitea Building Lab.

Conceptos Técnicos: Transformaciones vs Modelos

Aitea Building Lab trabaja con dos tipos principales de componentes para el procesamiento de datos:



Transformación

- Procesos de limpieza y preparación de datos
- Heredan de `MetaTransform`
- Dispone de `fit`, `fit_transform` y `transform`
- Generalmente no requieren memoria
- Se pueden conectar con otras piezas



Modelo

- Algoritmos que requieren entrenamiento
- Heredan de `MetaModel`
- Dispone de `fit`, `predict` y `fit_predict`
- Necesitan memoria tras un `fit`
- Elemento terminal, no se conecta con otras piezas

Ambos conceptos conforman la base de una tubería (pipeline) de procesamiento en Aitea Building Lab

Codificación Práctica de Analíticas

Implementación de Clases:

- Transformaciones: heredan de **MetaTransform**
- Modelos: heredan de **MetaModel**
- Ubicación: directorio **models_warehouse/**

Métodos obligatorios:

- **fit**: Prepara/entrena el modelo (retorna self)
- **transform/predict**: Procesa los datos (retorna DataFrame)
- **get_info**: Información del modelo y versión
- **get_all_attributes**: Atributos del modelo

```
# Ejemplo básico de transformación
class MiTransformacion(MetaTransform): def
    __init__(self, parametros):
        self.parametros = parametros
        self.result_dictionary = {}

    def fit(self, X, y=None):
        # Preparación
        return self
```


Pruebas y Versionado

Pruebas Locales

Uso de display.py con Streamlit para validación:

- Selector de librerías disponibles
- Configuración de fechas y ventanas temporales
- Ejecución de tests individuales o baterías aleatorias
- Visualización de resultados en gráficas o tablas

Gestión de Versiones

Implementación basada en fecha-hora:

```
LIBRARY_VERSION =  
datetime.datetime.now().strftime("%Y_%m_%d_%H_%M")
```

Permite mantener un registro claro y cronológico de cada versión desplegada.

Proceso recomendado para pruebas:

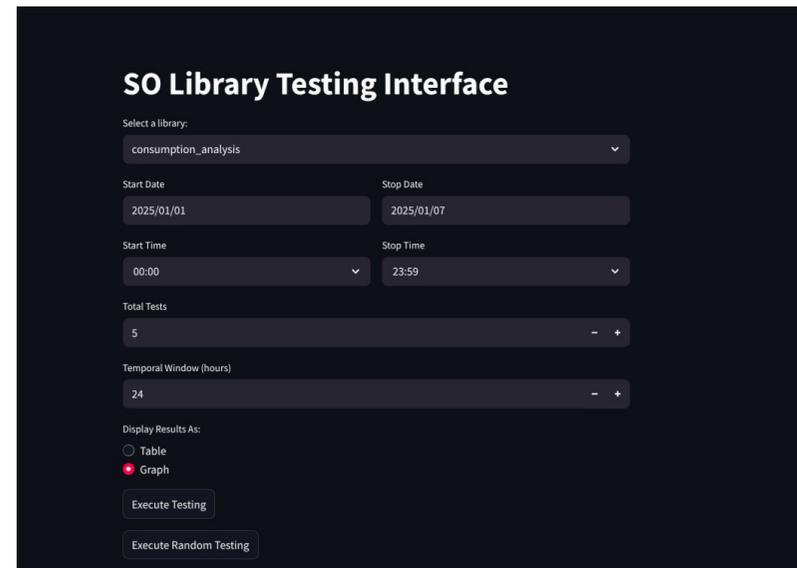
1. Desarrollo y depuración inicial con modo testing=True
2. Generación de librería con generate_so=True y testing=False

Pruebas y Versionado

Pruebas Locales (display)

Un entorno gráfico nos permite:

- Visualizar gráficamente los resultados.
- Visualizar tablas con resultados.
- Buscar rápidamente valores NaN y fallos en los cálculos.
- Testear para múltiples fechas y condiciones aleatorias



Conclusiones

Puntos clave a recordar:

- ◆ Las analíticas son algoritmos que extraen información valiosa de los datos de sensores
- ◆ El proceso de desarrollo sigue una metodología estructurada en 4 pasos
- ◆ La distinción entre Transformaciones y Modelos es fundamental en Aitea Building Lab
- ◆ El entrenamiento y pruebas son críticos para el éxito de las analíticas

Contacto para consultas técnicas:
info@aerin.es