gaia-x Hub España

RETO 1

Transferencia de Datos entre múltiples Espacios de Datos usando el Data Space Protocol para la construcción de modelos de IA

1a EDICIÓN DEL HACKATHON GAIA-X ESPAÑA

Impulsando la economía del dato española mediante la compartición y explotación de datos de forma confiable, soberana y segura

Fecha: 20/12/2024

Versión 01

PRIMERA EDICIÓN DEL HACKATHON GAIA-X ESPAÑA

2 y 3 de diciembre 2024 en Matadero Madrid

La Asociación **Gaia X España** y las entidades colaboradas **Universidad Politécnica de Madrid (UPM), Tecnalia, CTIC, AIRE NETWORKS y ARSYS**, celebraron la 1ª edición del Hackathon Gaia-X España, celebrado los días 2 y 3 de diciembre 2024 en Casa del Lector, Matadero en Madrid.

En el siguiente documento se presenta el **Reto 1** propuesto, el equipo y mentores que afrontaron el reto, la propuesta y enfoque del equipo participante, la presentación mostrada en el Hackathon y enlace al repositorio de código (si se encuentra disponible).

RETO 1

Transferencia de Datos entre múltiples Espacios de Datos usando el Data Space Protocol para la construcción de modelos de IA.

Propuesta Inicial

El reto UPM-Eunomia busca resolver los problemas existentes a nivel de transferencia entre diversos espacios de datos. Para ello, desde el proyecto Eunomia se ha desarrollado una implementación de código libre del Data Space Protocol llamada Rainbow. El objetivo principal del reto es que los participantes sean capaces de desplegar una infraestructura básica de un espacio de Datos MVDS (Minumum Viable Data Space), que luego irá evolucionando de forma incremental para acercarse en mayor medida a lo que sería un entorno real, permitiendo la transferencia de datos desde un Espacio de datos a por medio de los actores Proveedor y Consumidor. Una vez conseguida la transferencia a través de Rainbow, se debe utilizar el dataset transferido para generar un modelo de AI que

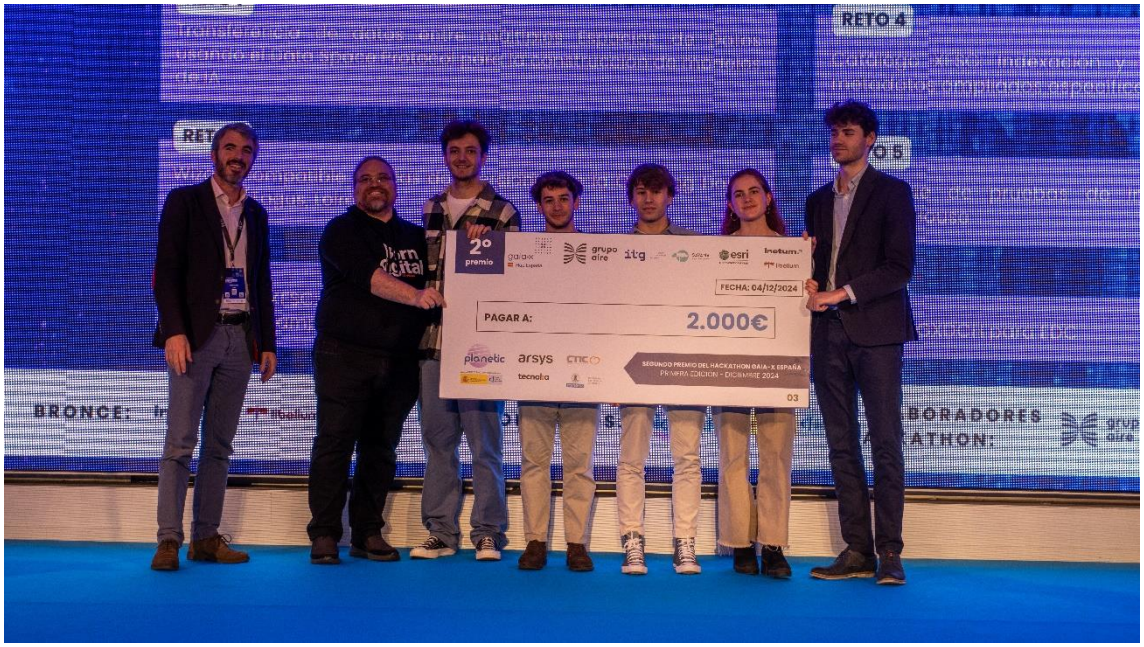
provea una solución para un problema específico del Espacio de datos de destino.

Mentores:
Carlos Aparicio (UPM), Rodrigo Menéndez (UPM), Javier Conde (UPM).
Responsable: Andrés Muñoz (UPM), Joaquín Salvachúa (UPM)
Entidad/es: Universidad Politécnica de Madrid (UPM).

Participantes – Equipo Hackathon

Jaime Alonso (UPM)
Jorge Suárez (UPM)
Jorge Rodríguez (UPM)
Marcos Rosado (UPM)
Cristina Rodríguez (UPM)
Daniel Alzueta (i2Cat)

Ganadores de un Premio de 2.000€



Enlace Presentación Final

Presentación Reto 1

Vídeo Demo Gaia-X Hackathon

Enlace Repositorio

https://github.com/alzcurda/ds-deployment

Propuesta y Enfoque del Equipo Participante

**Comentado por el propio equipo.*

Generar un flujo ETL mediante Data Space Connectors que implemente el DSP, que permita obtener datos de diferentes orígenes, almacene los datos en un lago de datos, los transforme, genere los datasets necesarios para su análisis y genere los modelos necesarios para su explotación. Adicionalmente se dispondrá de una UI para visualizar y explotar los resultados.
Como caso de uso queremos obtener un modelo que nos ayude a tomar decisiones a la hora de decidir cuántas viviendas y dónde podemos construirlas de forma que el impacto en la sociedad sea el mayor posible. Para dicho cálculo nos basamos en datos históricos de viviendas de la zona, servicios próximos y terrenos disponibles de la ciudad de Madrid.

Enfoque

Partimos del proyecto https://github.com/ging/ds-deployment , dónde disponemos de dos conectores implementados en Fiware y que se comunican utilizando el DSP. Lo primero que creamos son los datos origen. Datos origen Los datos de origen provienen de una serie de ficheros que hemos generado sintéticamente usando IA generativa. Los datos son:

Vivienda

Campo	Descripción
ID_Vivienda	Identificador único de la vivienda.
Coordenadas	Ubicación geográfica de la propiedad (latitud y longitud).
Fecha_Compra	Fecha de adquisición de la vivienda en formato de timestamp.
Fecha_Venta	Fecha de venta de la vivienda en formato de timestamp.
Precio_Compra (€)	Precio al que se adquirió la vivienda.
Precio_Venta (€)	Precio al que se vendió la vivienda.
Superficie (m²)	Tamaño de la vivienda en metros cuadrados.
Habitaciones	Número de habitaciones de la vivienda.
Baños	Número de baños de la vivienda.
Tipo	Tipo de vivienda (apartamento, piso, chalet, ático, etc.).
Distrito	Distrito de Madrid donde se encuentra la vivienda.

Servicios

Campo	Descripción
ID_Servicio	Identificador único para cada servicio.
Coordenadas	Ubicación geográfica del servicio (latitud y longitud).
Tipo_de_Servicio	Tipo de servicio ofrecido, como supermercado, colegio, hospital, estación de metro, centro comercial, parque, etc.

Terrenos libres

Campo	Descripción
ID_Terreno	Identificador único del terreno.
Nodos	Lista de coordenadas (latitud, longitud) que definen los vértices del terreno.
Superficie (m²)	Tamaño del terreno en metros cuadrados.
Precio (€)	Precio estimado del terreno en euros.

Rest Api Service

Para poder acceder a los datos de origen mediante los conectores creamos un servidor RestAPI usando FastAPI, con los siguientes endpoints (pueden ser accedidos mediante http://localhost:8000/docs):

Método	Endpoint	Descripción	Archivo asociado
GET	`/viviendas`	Devuelve el contenido del archivo con información de viviendas.	`Datos_Viviendas_Madrid.json`
GET	`/servicios`	Devuelve el contenido del archivo con información de servicios.	`Servicios_Madrid.json`
GET	`/terrenos`	Devuelve el contenido del archivo con información de terrenos libres.	`Terrenos_Libres_Poligonos_Madrid.json`

Rest Api Backend

Una vez creados los datos de origen, creamos la parte de backend encargada de dar de alta los catálogos de los conectores, registrar los assets e implementar toda la lógica y orquestación del flujo de trabajo. Para poder interactuar con el backend generamos otro servidor RestAPI con los siguientes endpoints:

Método	Endpoint	Descripción
GET	`/get_transfer_data`	Realiza el flujo de transferencia de datos, crea catálogos y acuerdos, descarga datos y los guarda como CSV.
GET	`/generate_dataset_viviendas`	Genera un dataset cruzando datos de viviendas y servicios mediante el módulo ETL.
GET	`/generate_dataset_terrenos`	Genera un dataset cruzando datos de terrenos y servicios mediante el módulo ETL.
GET	`/generate_viviendas_con_nota`	Genera un dataset de viviendas con una nota asignada utilizando un modelo de machine learning.
GET	`/get_viviendas`	Devuelve el dataset de viviendas con notas generado por el modelo de machine learning.
GET	`/get_terrenos`	Devuelve el dataset de terrenos y servicios generado mediante el módulo ETL.

Modelado IA

El modelo de Machine Learning que utilizamos es un Random Forest Regressor. Su objetivo es predecir un valor numérico que representa una "nota" para cada una de las viviendas. El funcionamiento es el siguiente:

1. Entrenamiento del modelo
   El modelo necesita aprender cómo las características de las viviendas (precio, tamaño, proximidad a servicios, etc.) están relacionadas con una calificación o nota. Este proceso se realiza en varias etapas:
   A. Los datos iniciales:
      a. Separte de un conjunto de datos en el que cada vivienda tiene:
         i. Características (o atributos): Precio de compra, número de habitaciones, cercanía a servicios, etc.
         ii. Etiqueta objetivo (target): Una nota ficticia generada aleatoriamente, que el modelo usa para aprender. La idea es que el valor de esta nota en próximas versiones no sea aleatorio si no que sea asignado a partir de otra referencia.
   B. Dividir los datos:

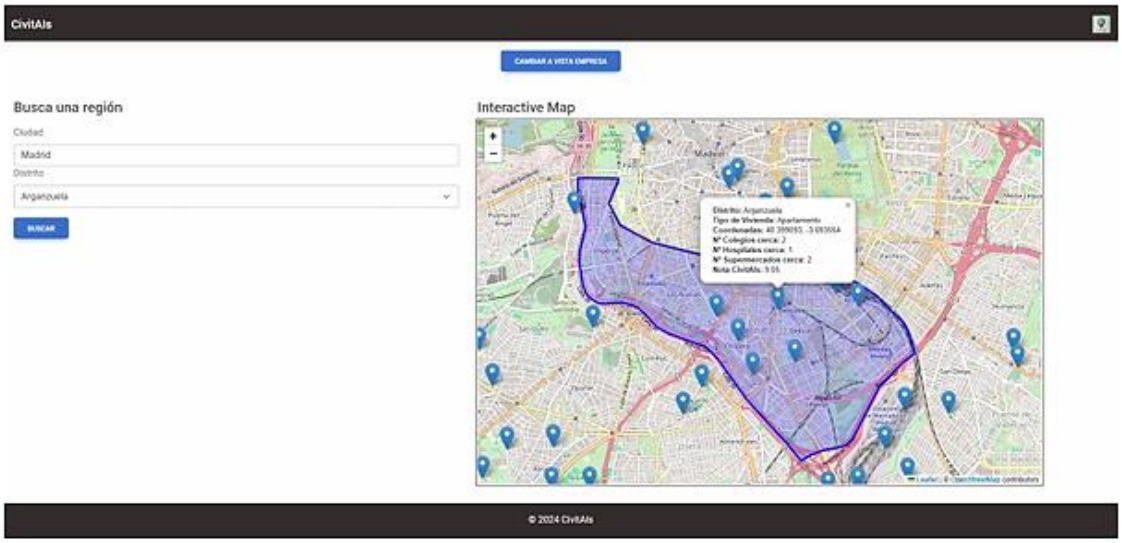
a. Los datos se dividen en dos partes:
    i. Entrenamiento: Usado para enseñarle al modelo.
    ii. Prueba: Usado para evaluar qué tan bien aprendió.

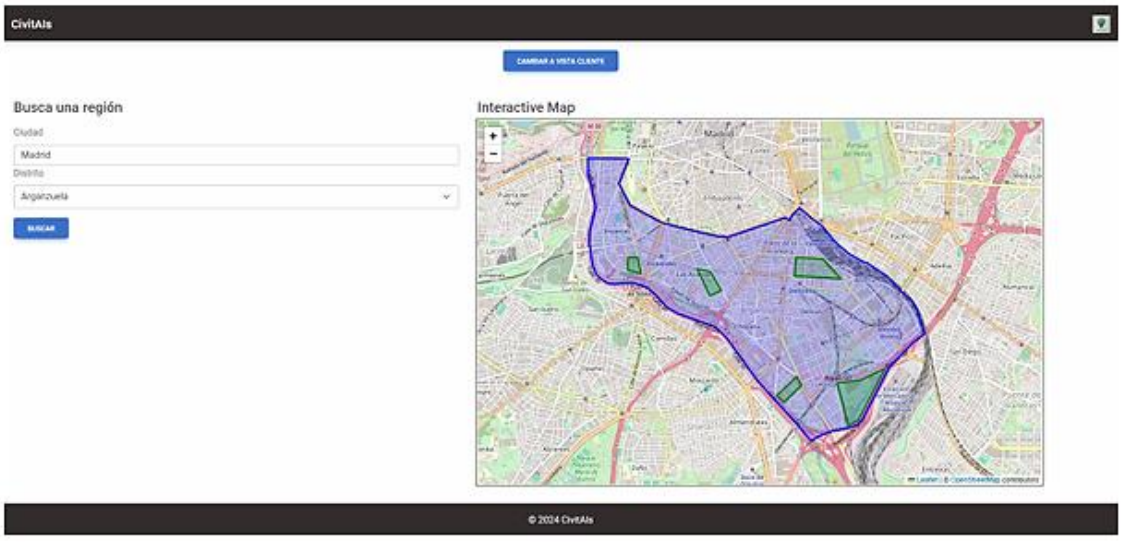
2. El Random Forest
   En nuestro caso lo que queríamos era poder obtener una nota de la idoneidad de la vivienda, y para ello usamos el tipo de Random Forest, ya que es útil tanto para problemas clasificatorios como de regresión.

3. Predicción
   Una vez que el modelo ha aprendido de los datos de entrenamiento:
    1. Se le pueden dar datos nuevos (características de una vivienda sin nota).
    2. Cada árbol en el bosque analiza las características y hace una predicción.
    3. El modelo combina estas predicciones para generar el valor final de la Nota.

UI

Nuestra interfaz es una herramienta que permite visualizar, en un mapa interactivo, diferentes regiones e inmuebles de una ciudad seleccionada. Los usuarios pueden explorar estas regiones y consultar diversas métricas que categorizan tanto los inmuebles como las regiones en las que se encuentran.


CivitAIs
CAMBIAR A VISTA EMPRESA
Busca una región
Ciudad
Madrid
Distrito
Arganzuela
BUSCAR
Interactive Map
© 2024 CivitAIs


CivitAIs
CAMBIAR A VISTA CLIENTE
Busca una región
Ciudad
Madrid
Distrito
Arganzuela
BUSCAR
Interactive Map
© 2024 CivitAIs