Pipeline de análisis de datos del cosmos en Microsoft Azure

Daniel García Algora
David García Guillén
Álvaro Tena Tamayo

Contenidos

Introducción

Este proyecto consiste en utilizar las herramientas de Microsoft Azure para llevar a cabo un análisis de datos por medio del despliegue de un pipeline completo en la nube. Para ello, se realiza un análisis y preprocesamiento de los datos en Azure Synapse Analytics, para después entrenar y evaluar el modelo en Azure Machine Learning. Como objetivo adicional, se conecta un cuadro de mando de PowerBI con Synapse Analytics para la visualización de los datos.

Se adjunta, además de enlaces a todos los enlaces relevantes, un vídeo demostración de los resultados.

Azure Synapse Analytics

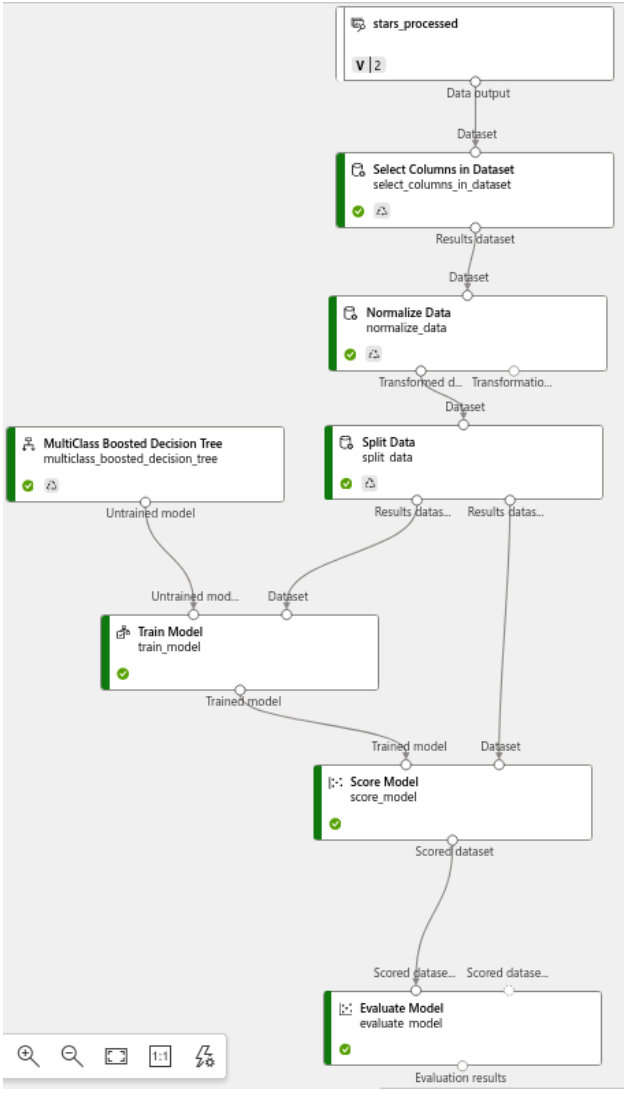
Para realizar la primera tarea de análisis y preprocesamiento de los datos, contenida en este *workspace*, se ha desplegado un entorno de *jupyter notebook*. Sobre este entorno, se ha desarrollado de forma programática la solución empleando *pySpark*, *SQL* y *python* puro. Para la ingesta inicial, se obtuvo el conjunto de datos stars.csv a través de una búsqueda en GitHub, lo cual facilita obtener su URL en formato *raw* para obtener los datos a través de una URL en el servicio de Azure. El tratamiento incluye tratamiento de valores nulos, establecimiento de tipos correctos, y limpieza de valores no-numéricos (para que los algoritmos puedan tratarlos). El conjunto de datos final se almacena por separado, para adaptar el modelo posteriormente.

El *notebook* stars_classification.ipynb con la limpieza de los datos puede encontrarse en este enlace y en el repositorio de la asignatura. El resultado se almacena en el fichero *stars_transformed_csv,* contenido en el *datalake*.

Azure Machine Learning

Esta segunda tarea consiste en definir un *pipeline* que permita entrenar y evaluar un modelo de aprendizaje automático. En este caso, para este dataset concreto, se busca predecir la clasificación estelar (*spectral_class*) de las estrellas contenidas en el conjunto de datos. Se trata de una medida que separa ciertos tipos de estrellas en función de sus características espectrales, como la emisión de radiación electromagnética.
Todos los experimentos realizados están contenidos en este *workspace*.

Empleando la herramienta *designer*, se ha definido el siguiente pipeline, que engloba la carga de los datos preprocesados del *datalake*, el entrenamiento del modelo y su evaluación:



La parte “genérica” del *pipeline* consiste en obtener los datos del *datalake*, normalizarlos y dividirlos en datos de entrenamiento y validación. Posteriormente, se entrena un modelo y se evalúa frente a los datos de validación.

Sobre este mismo *pipeline*, se han probado una serie de modelos, entre los cuales pueden encontrarse un árbol de decisión (experimento DecisionTreeTrain), un modelo de red de neuronas profunda (experimento DeepStarModel-Train) y un algoritmo de *boosting* (experimento BoostStarModel-Train-v2) con la finalidad de encontrar el que mejor se adapte a los datos.

Finalmente, se obtienen las siguientes métricas de evaluación (nótese que se mencionan por sus nombres en inglés *precision*, *recall* y *accuracy* para evitar confusiones):

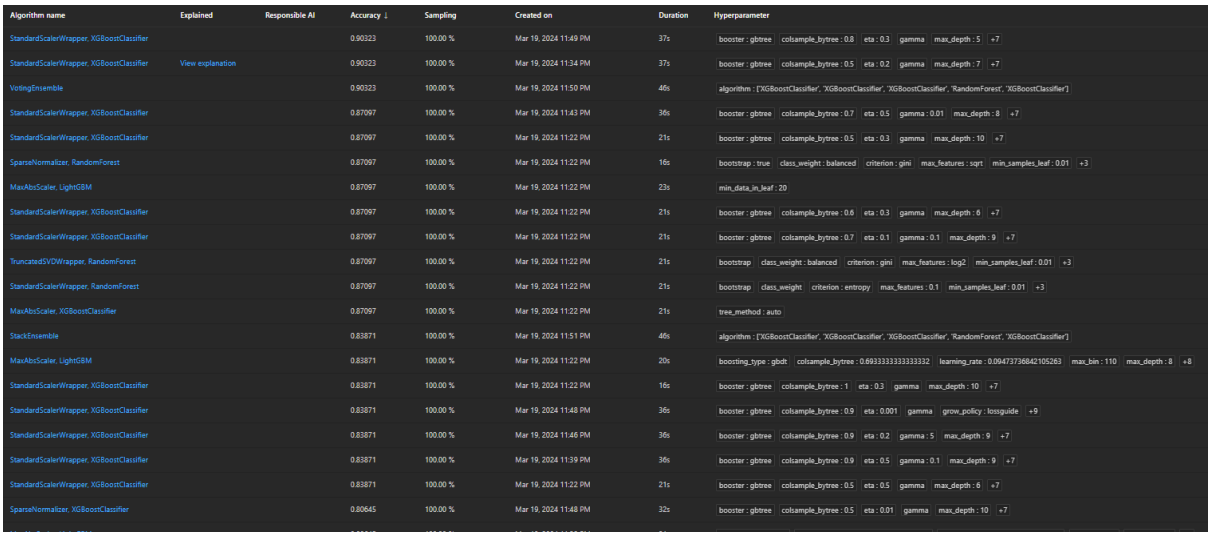
Modelo	Precision	Recall	Accuracy
Árbol de decisión	0.6607143	0.6626984	0.8333333
Red profunda	0.0625	0.1666667	0.375
Boosting	0.8524	0.8360	0.8542
Boosting + Stratified	0.9398148	0.9398148	0.9574468

Como conclusiones de estas pruebas manuales, se puede afirmar con seguridad que la red de neuronas profunda, como es de esperar, presenta un grave sobreajuste ya que el conjunto de datos cuenta únicamente con cerca de 300 líneas. No supera las métricas del árbol de decisión, que se ha implementado como punto de partida por ser el algoritmo más básico.
El mejor resultado lo obtiene el algoritmo de *boosting*, que, a pesar de su tendencia al sobreajuste, logra un resultado robusto.

Los resultados mejoran notablemente al marcar la casilla *stratified* a la hora de dividir los datos, ya que presentan un considerable desbalance en la variable objetivo.

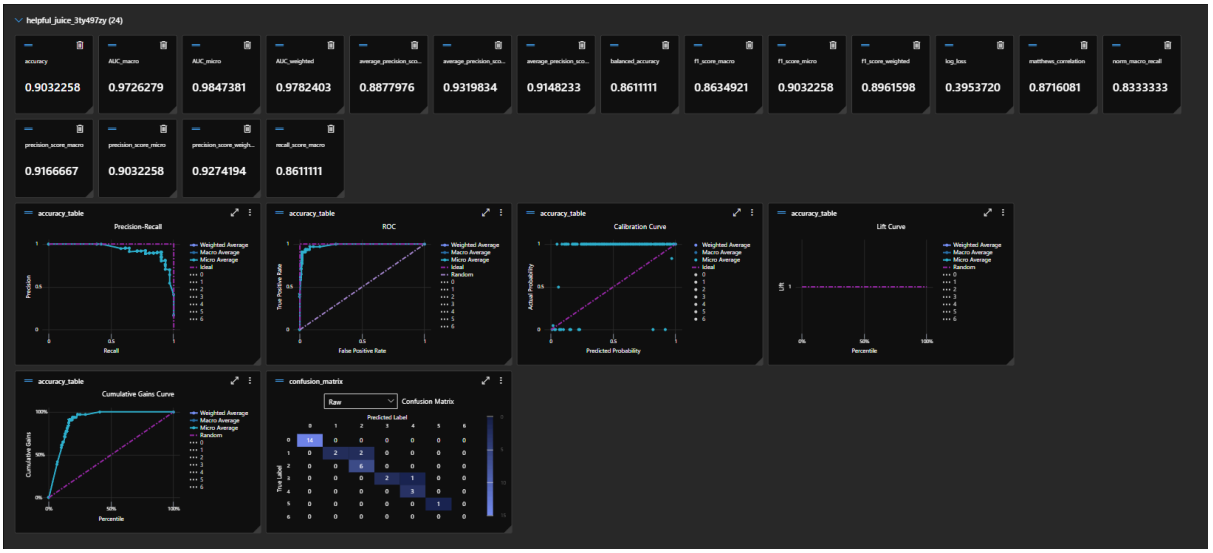
Pipeline de AutoML

Además del proceso manual descrito anteriormente, se ha desplegado un *pipeline* de AutoML para buscar arquitecturas que se ajusten a los datos. Se han ejecutado más de 40 pruebas:



Algorithm name	Explained	Responsible AI	Accuracy	Sampling	Created on	Duration	Hyperparameter
StandardScalerWrapper, XGBoostClassifier			0.90323	100.00 %	Mar 19, 2024 11:49 PM	37s	booster : gbtree colsample_bytree : 0.8 eta : 0.3 gamma max_depth : 5 +7
StandardScalerWrapper, XGBoostClassifier	View explanation		0.90323	100.00 %	Mar 19, 2024 11:34 PM	37s	booster : gbtree colsample_bytree : 0.5 eta : 0.2 gamma max_depth : 7 +7
VotingEnsemble			0.90323	100.00 %	Mar 19, 2024 11:50 PM	46s	algorithm : ['XGBoostClassifier', 'XGBoostClassifier', 'XGBoostClassifier', 'RandomForest', 'XGBoostClassifier']
StandardScalerWrapper, XGBoostClassifier			0.87097	100.00 %	Mar 19, 2024 11:43 PM	36s	booster : gbtree colsample_bytree : 0.7 eta : 0.5 gamma : 0.01 max_depth : 8 +7
StandardScalerWrapper, XGBoostClassifier			0.87097	100.00 %	Mar 19, 2024 11:22 PM	21s	booster : gbtree colsample_bytree : 0.5 eta : 0.3 gamma max_depth : 10 +7
SparseNormalizer, RandomForest			0.87097	100.00 %	Mar 19, 2024 11:22 PM	16s	bootstrap : true class_weight : balanced criterion : gini max_features : sqrt min_samples_leaf : 0.01 +3
MaxAbsScaler, LightGBM			0.87097	100.00 %	Mar 19, 2024 11:22 PM	23s	min_data_in_leaf : 20
StandardScalerWrapper, XGBoostClassifier			0.87097	100.00 %	Mar 19, 2024 11:22 PM	21s	booster : gbtree colsample_bytree : 0.6 eta : 0.3 gamma max_depth : 6 +7
StandardScalerWrapper, XGBoostClassifier			0.87097	100.00 %	Mar 19, 2024 11:22 PM	21s	booster : gbtree colsample_bytree : 0.7 eta : 0.1 gamma : 0.1 max_depth : 9 +7
TruncatedSVDWrapper, RandomForest			0.87097	100.00 %	Mar 19, 2024 11:22 PM	21s	bootstrap class_weight : balanced criterion : gini max_features : log2 min_samples_leaf : 0.01 +3
StandardScalerWrapper, RandomForest			0.87097	100.00 %	Mar 19, 2024 11:22 PM	21s	bootstrap class_weight criterion : entropy max_features : 0.1 min_samples_leaf : 0.01 +3
MaxAbsScaler, XGBoostClassifier			0.87097	100.00 %	Mar 19, 2024 11:22 PM	21s	tree_method : auto
StackEnsemble			0.83871	100.00 %	Mar 19, 2024 11:51 PM	46s	algorithm : ['XGBoostClassifier', 'XGBoostClassifier', 'XGBoostClassifier', 'RandomForest', 'XGBoostClassifier']
MaxAbsScaler, LightGBM			0.83871	100.00 %	Mar 19, 2024 11:22 PM	20s	boosting_type : gbdt colsample_bytree : 0.6933333333333332 learning_rate : 0.09473736842105263 max_bin : 110 max_depth : 8 +8
StandardScalerWrapper, XGBoostClassifier			0.83871	100.00 %	Mar 19, 2024 11:22 PM	16s	booster : gbtree colsample_bytree : 1 eta : 0.3 gamma max_depth : 10 +7
StandardScalerWrapper, XGBoostClassifier			0.83871	100.00 %	Mar 19, 2024 11:48 PM	36s	booster : gbtree colsample_bytree : 0.9 eta : 0.001 gamma grow_policy : lossguide +9
StandardScalerWrapper, XGBoostClassifier			0.83871	100.00 %	Mar 19, 2024 11:46 PM	36s	booster : gbtree colsample_bytree : 0.9 eta : 0.2 gamma : 5 max_depth : 9 +7
StandardScalerWrapper, XGBoostClassifier			0.83871	100.00 %	Mar 19, 2024 11:39 PM	36s	booster : gbtree colsample_bytree : 0.9 eta : 0.5 gamma : 0.1 max_depth : 9 +7
StandardScalerWrapper, XGBoostClassifier			0.83871	100.00 %	Mar 19, 2024 11:22 PM	21s	booster : gbtree colsample_bytree : 0.5 eta : 0.5 gamma max_depth : 6 +7
SparseNormalizer, XGBoostClassifier			0.80645	100.00 %	Mar 19, 2024 11:48 PM	32s	booster : gbtree colsample_bytree : 0.5 eta : 0.01 gamma max_depth : 10 +7

El modelo con mejor *accuracy* encontrado es XGBoost, con un 90%, como se muestra en la figura:



Análisis de costes

En este apartado se encontrará el análisis de los costes, desglosando dónde se han gastado los créditos. Cómo se puede observar, el coste total del workspace más todos los experimentos es de **18.02€**. Este costo total se ha repartido entre varios recursos como se puede mostrar en la imágen. Lo que más ha consumido de crédito es el ***Synapse workspace***, debido a que la creación del workspace son **5USD** for each TB (Figura 2) plus **3.48USD** (Figura 3) para cada hora. Por último, el siguiente mayor coste es el que han causado los experimentos ejecutados mediante Azure Machine Learning. Con un **1.72€** por **5** experimentos completados, **2** erróneos, **1** cancelado y **1** *AutoML* mencionados en los apartados anteriores, teniendo en cuenta que son **0.32USD** la hora de ejecución (Figura 4).

Coste de Azure desglosado:

- Ejecución ejecución EDA notebook
    - Apache Spark Pool = 8h 12min 15s
        - 12 vcores usados + 4 cores del driver = 16 vcores
        - número de nodos = 16 cores / 4 cores * 1 nodo = 4 nodos
        - Tarifa de Azure -> 0.58USD/h de un 1 nodo
        - 4nodos * 0.58USD/h * (8 * 60 + 12) / 60 = 19,02 USD
- Azure ML
        - 45 minutos en total de trabajos
        - 3 esperas * 15min/espera = 45min
        - 0,32USD/h * (45min + 45min)/60 = 0,48 USD

- AzureML workspace, 1.63USD

Nos desviamos por un margen de 1.38 USD debido a que se borraron algunos procesos que, aunque no llegaron a completar su ejecución, se ejecutaron durante unos segundos.

Figura 1: Synapse

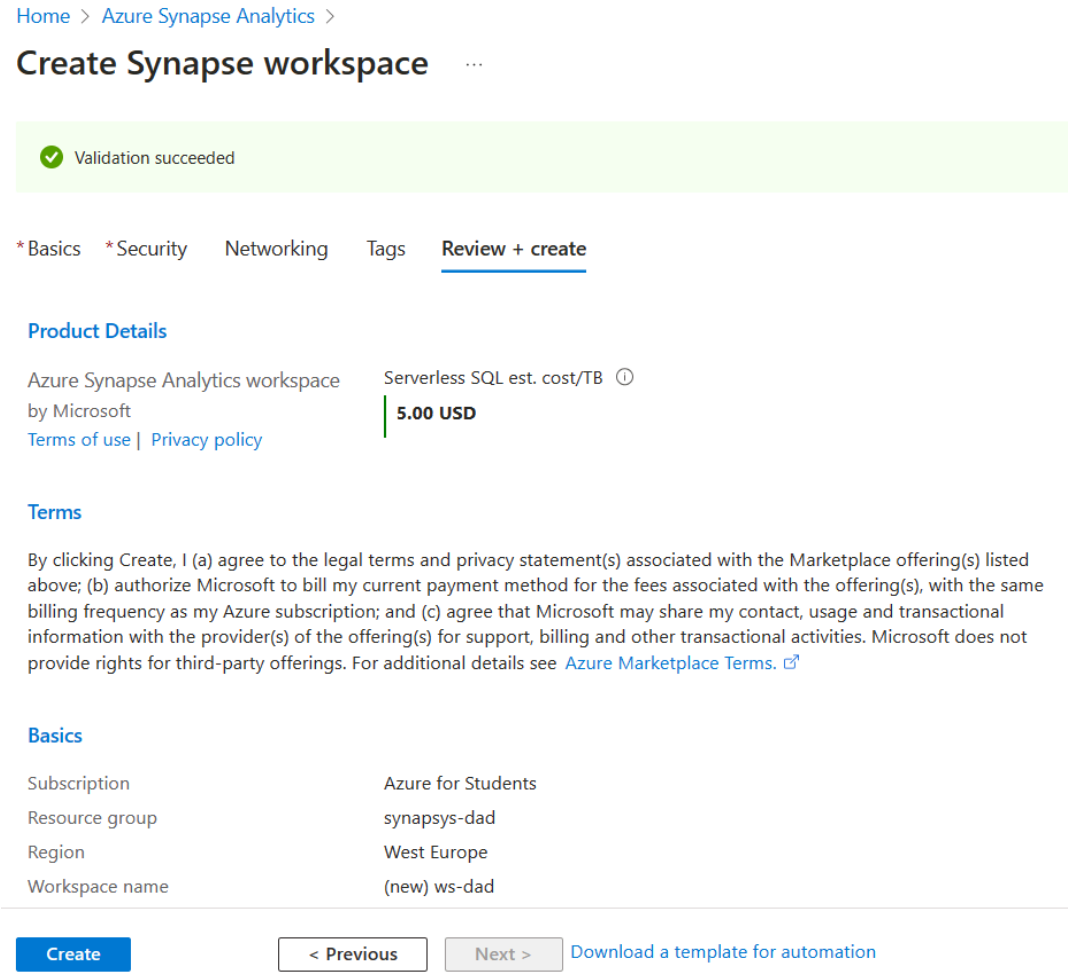


Figura 2: Apache Spark pool creation

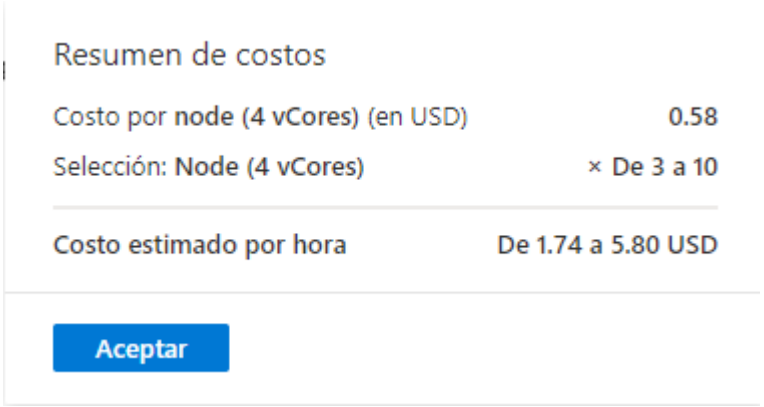


Figura 3. AzureML compute:

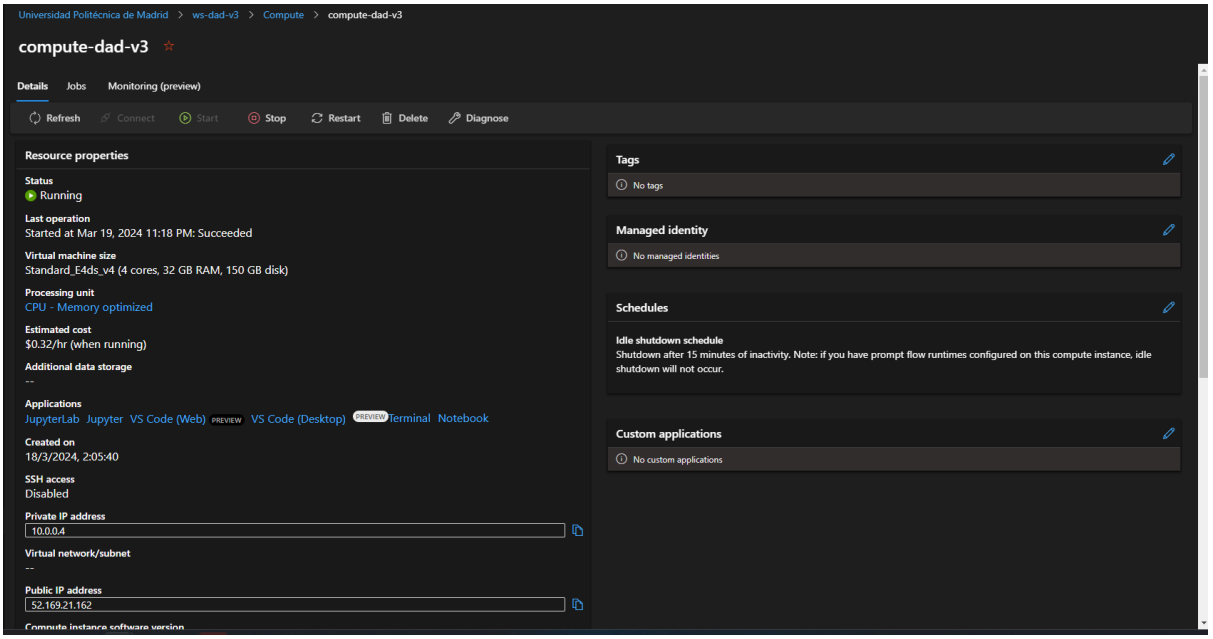
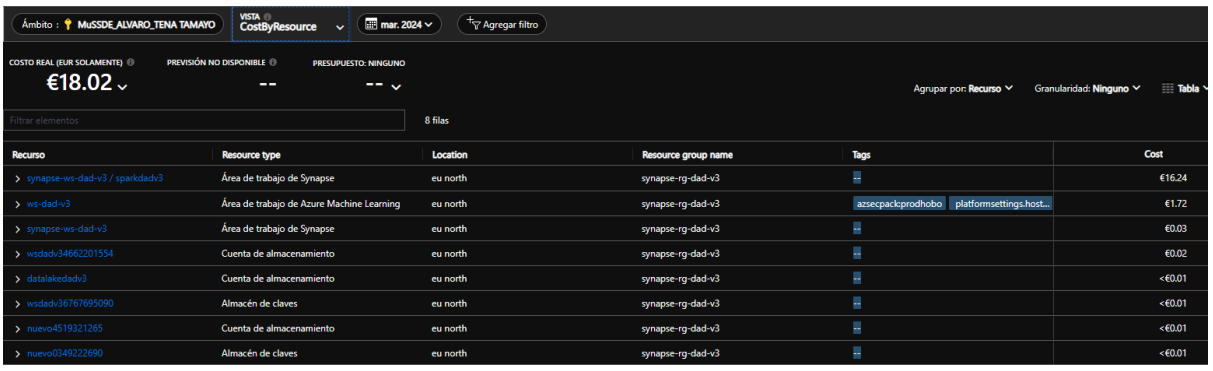
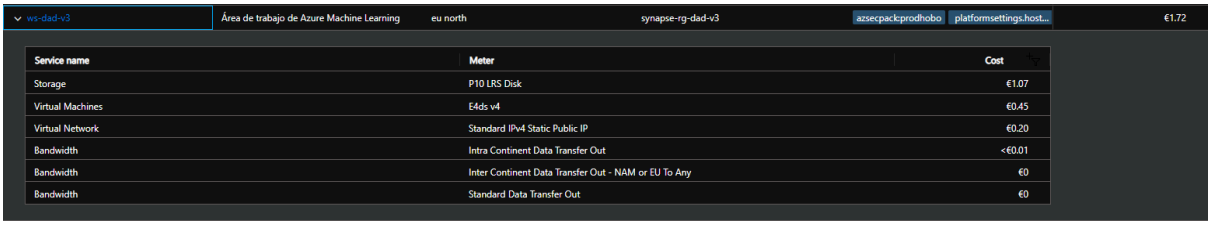


Figura 4. Desglose de costes:





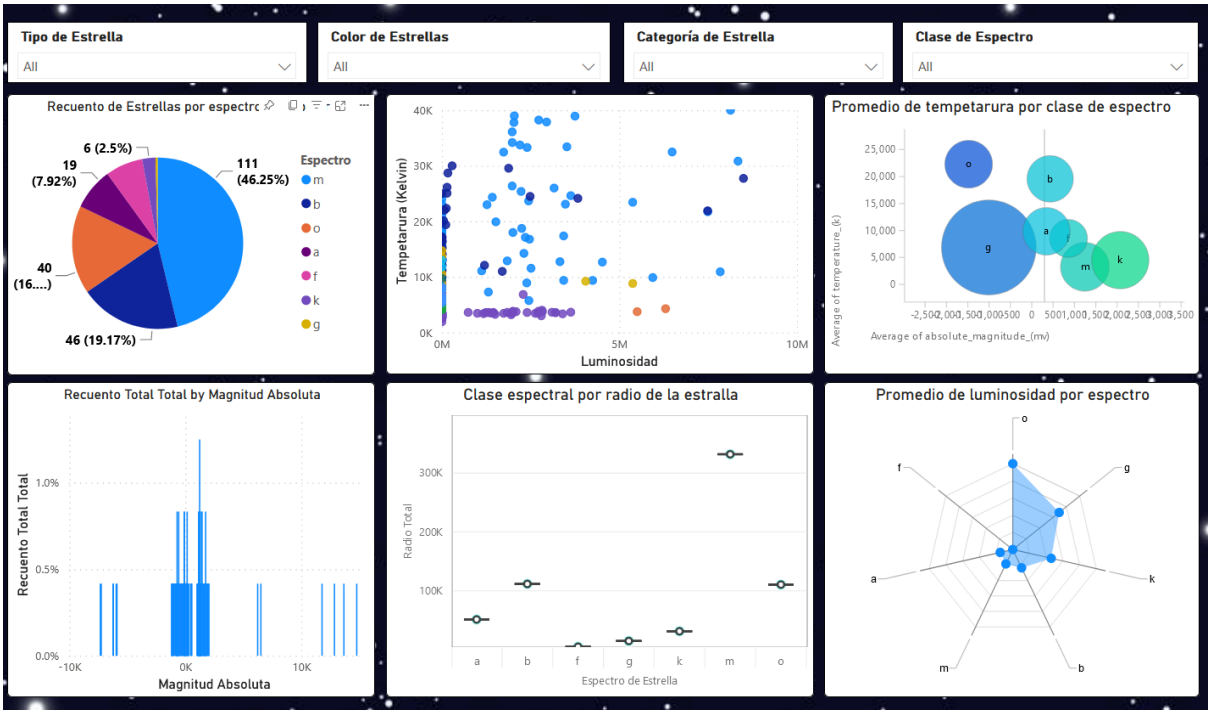
Service name	Meter	Cost
Storage	P10 LRS Disk	€1.07
Virtual Machines	E4ds v4	€0.45
Virtual Network	Standard IPv4 Static Public IP	€0.20
Bandwidth	Intra Continent Data Transfer Out	<€0.01
Bandwidth	Inter Continent Data Transfer Out - NAM or EU To Any	€0
Bandwidth	Standard Data Transfer Out	€0

Anexo: PowerBI

Se ha llevado a cabo, como objetivo adicional, un cuadro de mando en PowerBI, que permite visualizar el comportamiento y la distribución de las características de los cuerpos celestes en función de su tipo, color, categoría y clase de espectro. Puede encontrarse en el [repositorio de la asignatura](#).

La figura ilustra un ejemplo del funcionamiento de PowerBi:

Figura 5. Cuadro de mando en PowerBi:



Anexo: Vídeo demostración

Se incluye, en el siguiente enlace, un vídeo con una demostración del funcionamiento de todo el trabajo desplegado tal como se describe anteriormente:

https://upm365-my.sharepoint.com/:v:/g/personal/d_galgora_alumnos_upm_es/EW77Xgf8Fr1Okb5viuHc7BkB8UheIPRpAH0CjeHfJs39OA?e=bmlSC8

Este enlace puede encontrarse también en el fichero README.md del repositorio de la asignatura.