

Azure - Apple Quality

Pablo Torre e Ignacio Jiménez

Descripción General

En esta práctica vamos a resolver un problema de clasificación empleando Azure Synapse y Azure ML Studio. El problema consiste en determinar si una manzana tiene o no calidad suficiente a partir de una serie de variables:

- **Size:** Tamaño de la manzana
- **Weight:** Peso de la manzana
- **Sweetness:** Dulzura de la manzana
- **Crunchiness:** Textura, indicando lo crujiente que es cada manzana
- **Juiciness:** Nivel de jugosidad de la manzana
- **Ripeness:** Estado de madurez de la manzana
- **Acidity:** Nivel de acidez de la manzana
- **Quality:** Calidad de la manzana (el la etiqueta a predecir, binaria)

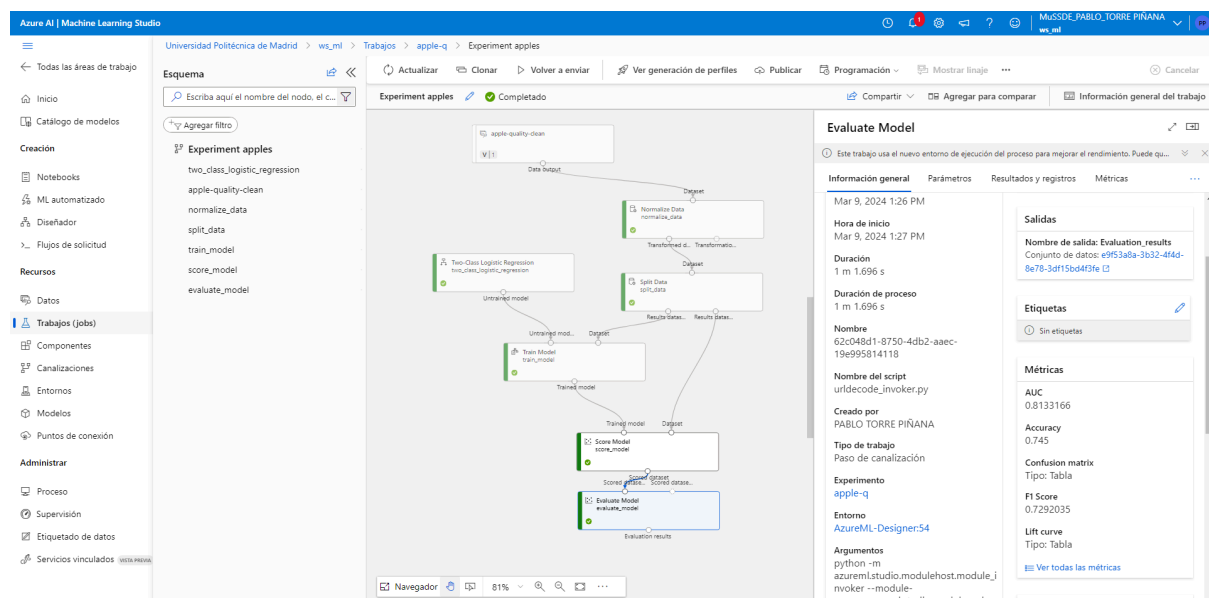
El dataset se puede encontrar en el siguiente [link](#)

Desarrollo

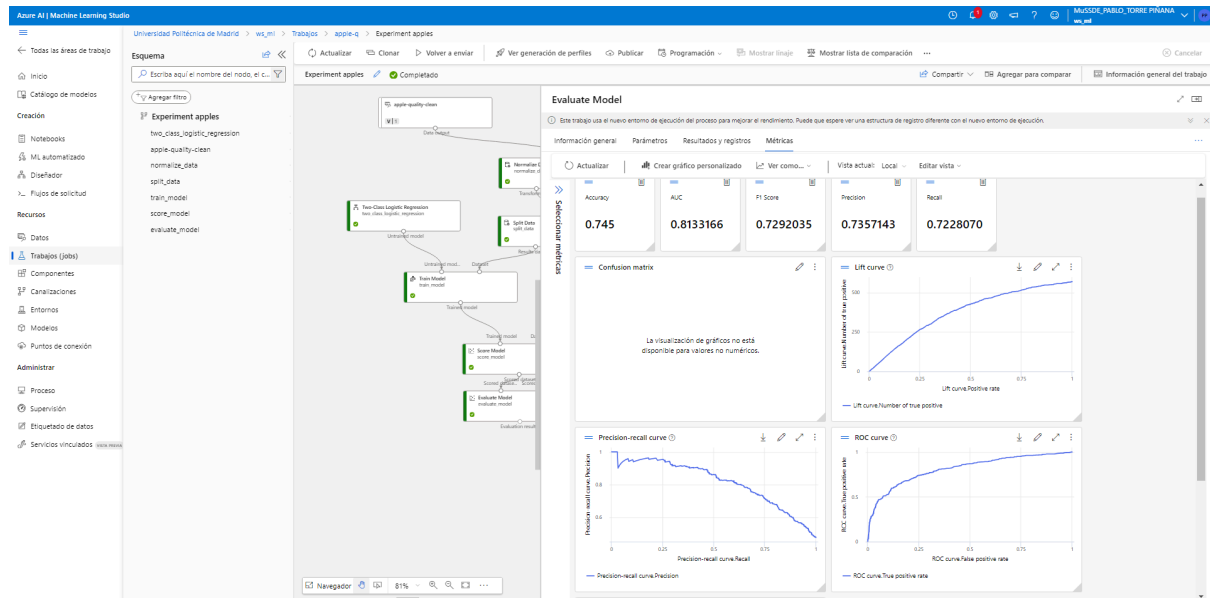
Hemos estructurado el desarrollo en dos fases: la primera se corresponde a la exploración de los datos y a la limpieza del dataset con *Synapse*, y la segunda es el entrenamiento de modelos de ML para predecir la calidad.

El cuaderno realizado en *Synapse* se puede encontrar en nuestro Github, el código está basado en el siguiente cuaderno de [kaggle](#). Para hacer la carga de datos, subimos el csv a nuestro github y lo cargamos a nuestro *Data Lake*. El csv final que obtenemos después de la visualización, análisis, limpieza y normalización lo guardamos en nuestro *Data Lake* y lo subimos a github para utilizarlo posteriormente en las tareas de predicción usando Azure ML Studio

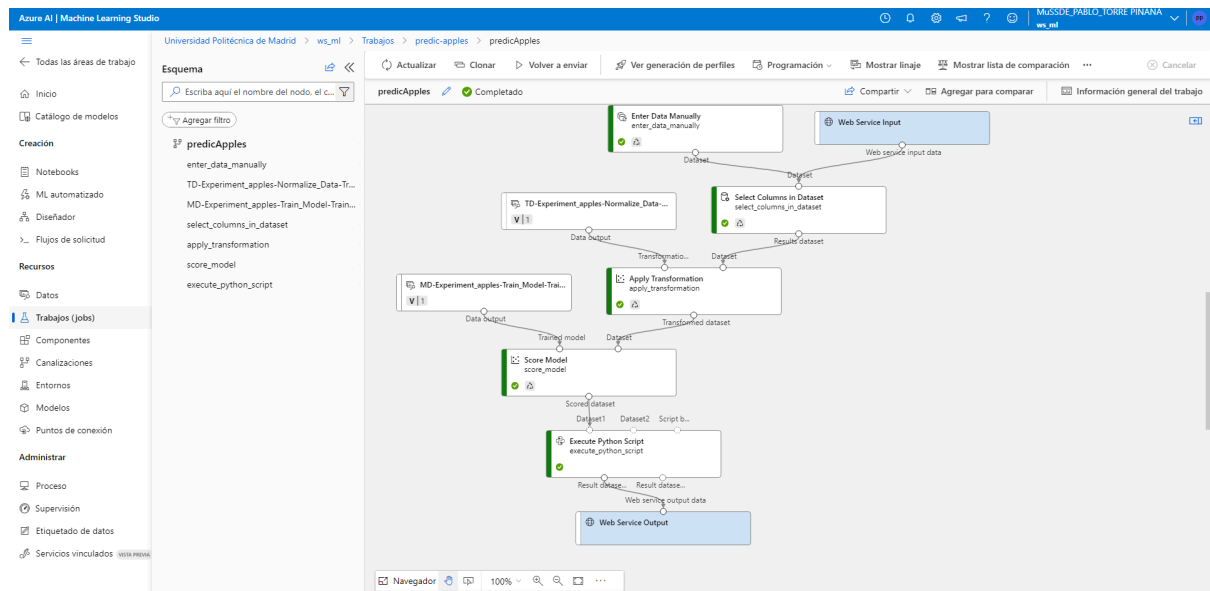
Para la segunda parte, hemos empleado dos métodos: *Designer* y *Automated ML*. Para el *Designer* hemos creado un pipeline con el siguiente flujo de operaciones:

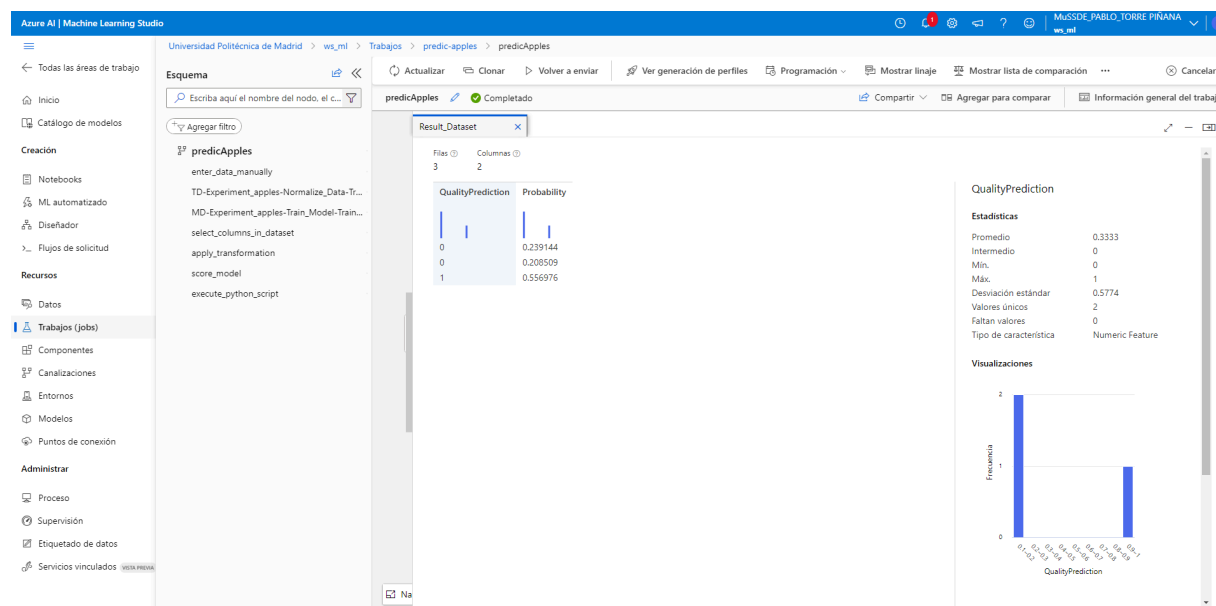


Cogemos los datos de nuestro dataset limpio, los normalizamos y hacemos un split para entrenar nuestro modelo de regresión de dos clases. Finalmente los evaluamos para ver la precisión de nuestro modelo. En la siguiente foto se puede ver un resumen de los resultados obtenidos:



Y, por último, creamos un Pipeline para realizar inferencia sobre el modelo entrenado. Lo probamos con tres ejemplos y vemos que el resultado es el esperado.





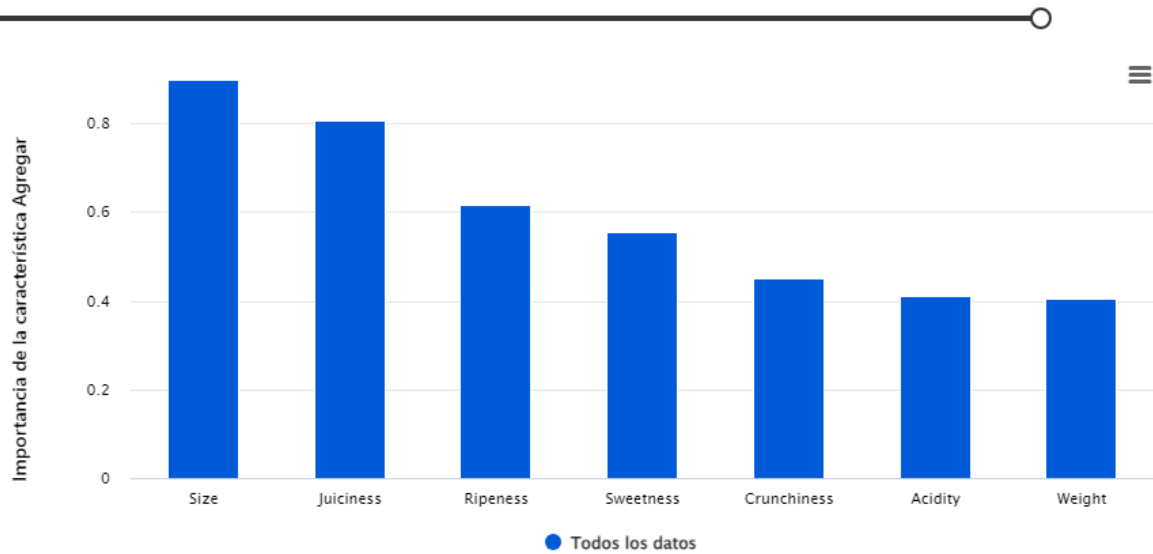
Con ML automatizado, se han probado alrededor de 75 modelos distintos. La ejecución ha tardado un total de 1h y 14 minutos. Los mejores resultados se pueden ver a continuación.

The screenshot shows the 'automatic_apples' workflow in Azure ML Studio. The 'Modelos y trabajos secundarios' (Models and secondary jobs) tab is selected, displaying a table of trained models. The table has the following columns: 'Nombre del algoritmo' (Algorithm name), 'Explicado' (Explained), 'IA responsable' (Responsible AI), 'Valor ponderado de AUC' (Weighted AUC value), 'Muestreo' (Sampling), 'Fecha de creación' (Creation date), 'Duración' (Duration), and 'Hiperparámetro' (Hyperparameter).

Nombre del algoritmo	Explicado	IA responsable	Valor ponderado de AUC	Muestreo	Fecha de creación	Duración	Hiperparámetro
VotingEnsemble	Ver explicación		0.97342	100.00 %	Mar 17, 2024 11:55 AM	1 m 10 s	algorithm : ['SVM', 'SVM', 'SVM', ...
StandardScalerWrapper. SVM			0.96828	100.00 %	Mar 17, 2024 11:25 AM	46 s	C : 51.79474679231202 class, ...
RobustScaler. SVM			0.96789	100.00 %	Mar 17, 2024 11:32 AM	45 s	C : 35.564803062231285 class, ...
StandardScalerWrapper. SVM			0.96789	100.00 %	Mar 17, 2024 11:07 AM	45 s	C : 11.513953993264458 class, ...
MaxAbsScaler. SVM			0.96743	100.00 %	Mar 17, 2024 11:10 AM	44 s	C : 11.513953993264458 class, ...
RobustScaler. SVM			0.96741	100.00 %	Mar 17, 2024 11:39 AM	45 s	C : 16.768329368110066 class, ...
RobustScaler. SVM			0.96724	100.00 %	Mar 17, 2024 11:47 AM	47 s	C : 75.43120063354607 class, ...
RobustScaler. SVM			0.96723	100.00 %	Mar 17, 2024 11:05 AM	47 s	C : 75.43120063354607 class, ...
RobustScaler. SVM			0.96675	100.00 %	Mar 17, 2024 11:31 AM	45 s	C : 5.428675439323859 class, ...
RobustScaler. SVM			0.96636	100.00 %	Mar 17, 2024 11:19 AM	51 s	C : 159.98587196060572 class, ...
RobustScaler. SVM			0.96508	100.00 %	Mar 17, 2024 11:35 AM	54 s	C : 339.3221771895323 class, ...
RobustScaler. SVM			0.96508	100.00 %	Mar 17, 2024 11:23 AM	53 s	C : 339.3221771895323 class, ...

Podemos acceder a un reporte muy detallado para la mejor de las soluciones. En las siguientes gráficas podemos ver cuales son las características más importantes según el mejor modelo y un pequeño análisis del rendimiento del mismo con un tamaño de muestra de 4000 filas. Como se puede ver obtuvimos una F1-score de 0.985, que es un valor muy alto para esta métrica.

Principales características de 7 por su importancia



Tamaño de muestra 4000

Precisión: 0,985

Precisión: 0,984

Coincidencia: 0,986

Puntuación F1: 0,985

Tasa de falsos positivos:

0,016

Tasa de falsos negativos:

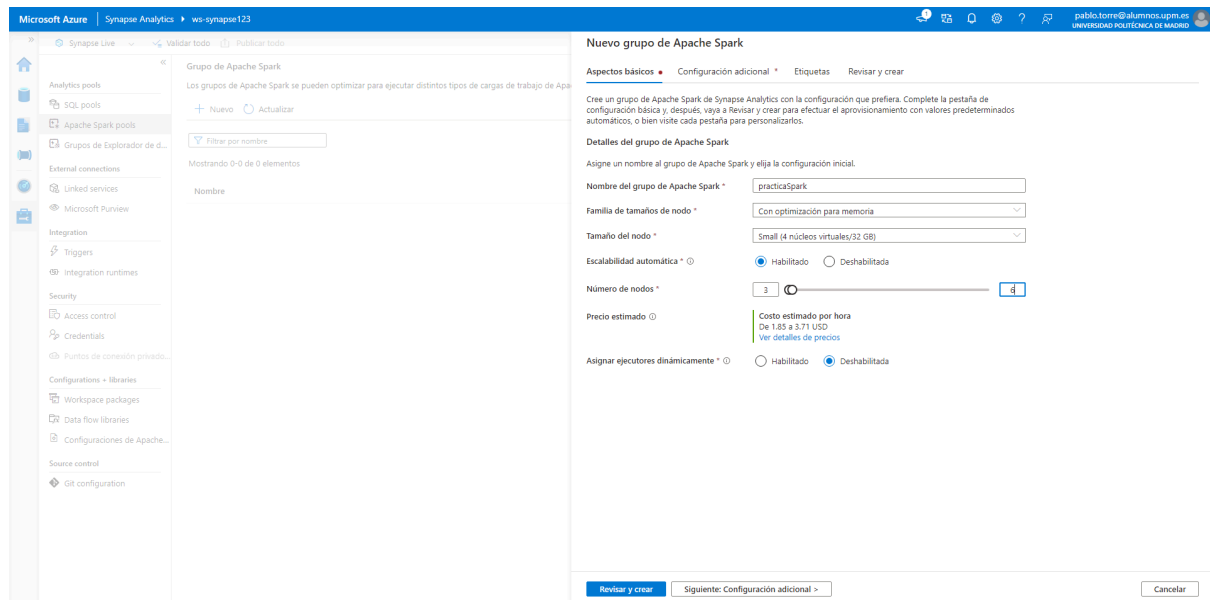
0,014

Tasa de selección: 0,5

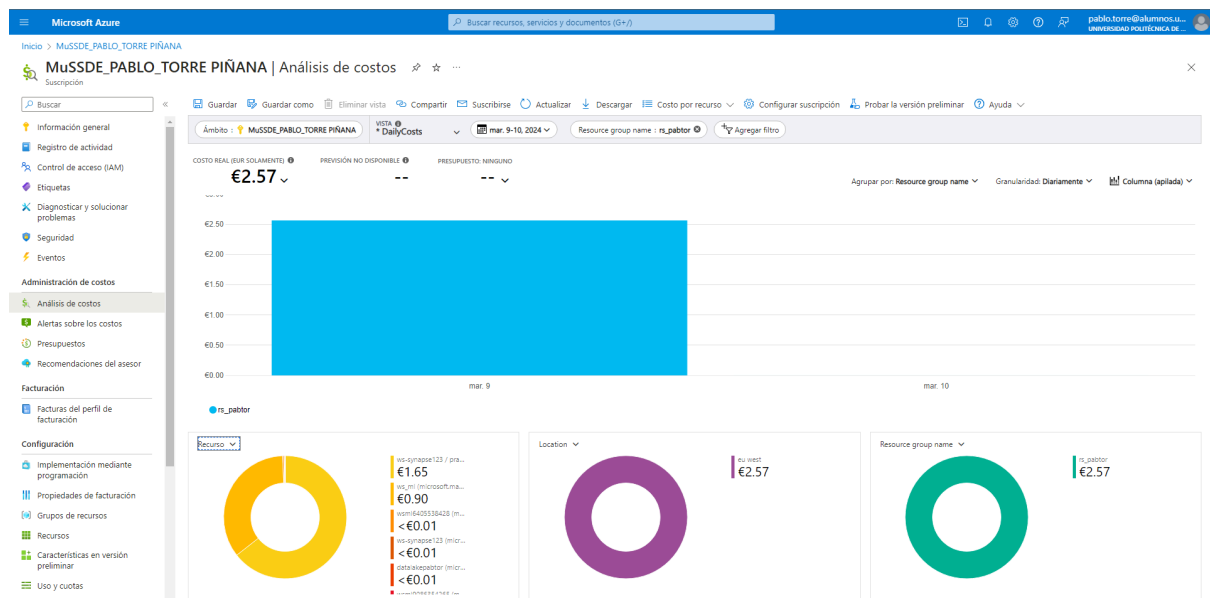
Resumen costes

Synapse

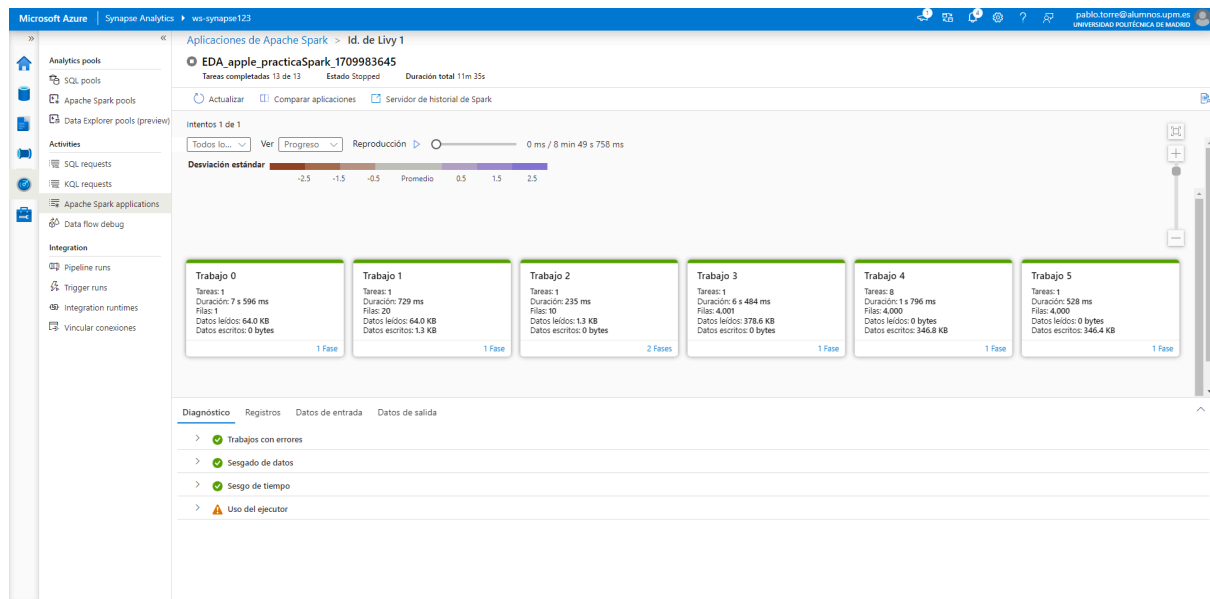
La configuración empleada tiene un costo estimado por hora de 1.85 a 3,71\$/h.



Los costes totales se pueden ver en la siguiente gráfica:



Se puede apreciar que de Synapse son 1.65€. Este coste se refiere al total de las ejecuciones de todas las pruebas que hicimos (unos 45-50 mins). La ejecución final fue tan solo de 11 mins, cómo se puede ver a continuación:



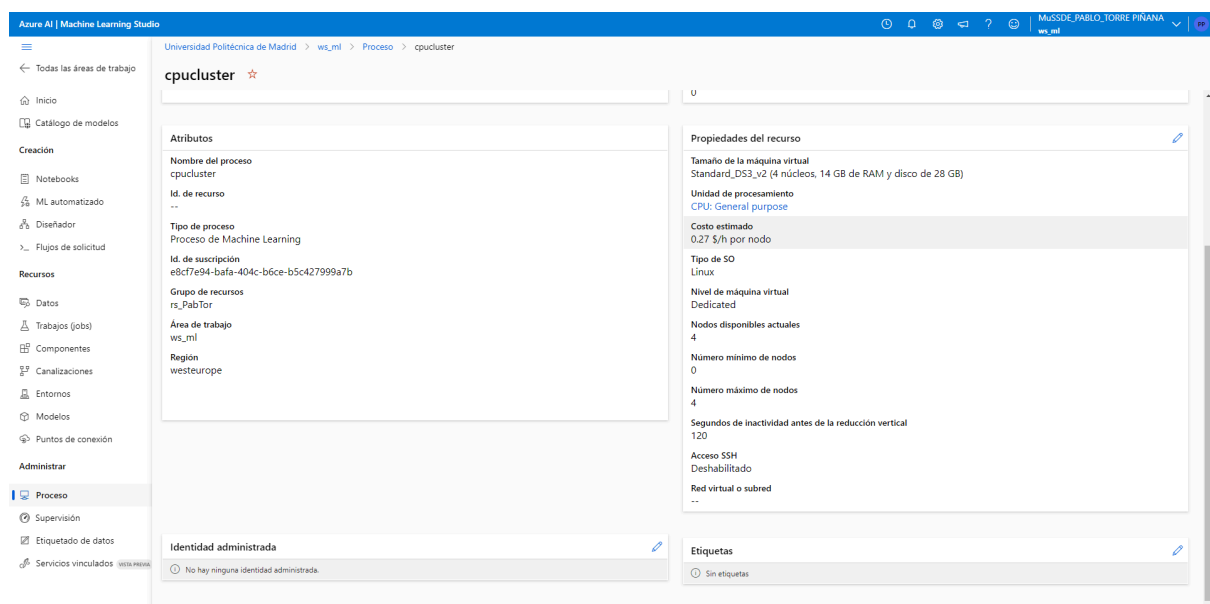
[2 executors*4vcores + 1 driver*4Vcores = 12Vcores]

(12/60)hours * 12vCores = 2.4 vCore hours

2.4 vCore hours * 0.147 = 0.3528€ + 0.01€ (datalake)

Designer

El entrenamiento del clasificador con Designer nos costó un total de 0.90€, como se puede ver en la imagen con el desglose por recurso del día, empleando la siguiente máquina (estuvo encendida desde el principio del día e hicimos varias pruebas previas a la final)



ML Automatizado

El entrenamiento de los distintos modelos utilizando el ML Automatizado nos costó 0.70€. Este proceso lo hicimos “serverless” de manera que no tuvimos que crear ningún cluster de proceso y solo pagamos por uso.

The screenshot shows the 'Envío de un trabajo de ML automatizado' (Send an automated ML job) configuration page in the Azure ML interface. The left sidebar contains navigation options like 'Inicio', 'Catálogo de modelos', 'Creación', 'ML automatizado', 'Diseñador', 'Flujos de solicitud', 'Recursos', 'Datos', 'Trabajos (jobs)', 'Componentes', 'Canalizaciones', 'Entornos', 'Modelos', 'Puntos de conexión', 'Administrar', 'Proceso', 'Supervisión', 'Etiquetado de datos', and 'Servicios vinculados'. The main area is titled 'Envío de un trabajo de ML automatizado' and includes a 'VISTA PREVIA' (Preview) button. The configuration steps are: 'Método de entrenamiento', 'Configuración básica', 'Tipo de tarea y datos', 'Configuración de tarea', 'Proceso', and 'Revisión'. The 'Proceso' step is selected, showing options for 'Tipo de máquina virtual' (CPU or GPU), 'Nivel de máquina virtual' (Dedicado or Prioridad baja), and 'Tamaño de la máquina virtual' (Standard_DS3_v2). The 'Número de instancias' is set to 1. The bottom of the page has 'Atrás', 'Siguiente', and 'Cancelar' buttons.

En la siguiente gráfica se pueden ver los costes por recurso del ML automatizado.

