Se pensó en la utilización del recurso de machine learning de Azure por las posibilidades para mlops y por la familiaridad que se le tiene. La idea es que, a partir de ciertos archivos subidos al blob storage (por un proceso de ETL por ejemplo) se cree una pipeline de entrenamiento que se activará cada 7 días. Los archivos necesarios a subirse son: el correspondientes a los datos de entrenamiento tal como se presentan inicialmente en el desafío, un archivo correspondiente a las librerías necesarias para el preprocesado de los datos y el entrenamiento, uno correspondiente al scoring file que utilizará el modelo deployado para predecir, y dos archivos de texto que se corresponden con listas de productos y de tiendas útiles en ciertas partes del preprocesado y que surgen del análisis exploratorio de los datos.

La pipeline de entrenamiento se realizó a partir de la clase Pipeline de azure ml que permite generar una pipeline con pasos constituidos por archivos .py que ejecutan cada parte del proceso. A su vez el objeto Pipeline puede luego publicarse y asociársele un Schedule. Los pasos de la pipeline en orden son: pre-procesado y feat eng, training, evaluación, registro. Una vez corrida, y en caso de que registre un nuevo modelo, se activará una pipeline que consta de un único paso el cual deploya el modelo. Una vez deployado se puede obtener la url asociada a través de la interfaz de usuario de azure ml o bien buscar al servicio por su nombre y llamarlo con los datos con los que se quiere predecir. Esto último se realiza a través de un script testing.py el cual recibe como argumento el path de un json con las features en estado crudo. Los pasos de las pipelines están gobernados por scripts cuyos nombres se relacionan con la tarea asociada.

Para un orquestamiento más robusto se puede utilizar Airflow o Azure DevOps para crear y ejecutar las pipelines, las cuales podrán usar los mismos scripts provistos.

Subiéndose la carpeta provista a una instancia de computo en azure ml y parados dentro de ella se ejecuta pipeline\_creation.py. Este archivo deployará las dos pipelines (de entrenamiento y de deployado del modelo). Previamente debieran estar los archivos mencionados en una carpeta con el nombre BigMart, asociado en este caso al dataset y al problema.

Directorio sugerido en default blob storage (el scoring\_file debe seguir los lineamientos requeridos por azure ml para un scoring scritp, debe tener dos funciones init() donde se carga el modelo y run() donde se predice):

**BigMart**

----**raw**

------**data**

--------Train\_BigMart.csv

--------Test\_BigMart.csv

------dependencies\_file.yml

------scoring\_file.py

----**txts**

------prod\_no\_aplica.txt

------tiendas.txt

El sript testing.py puede correrse ahora pasándole como argumento (--path\_to\_test) la dirección del .json y debería arrojar la predicción.