PROYECTO FINAL BTC IA y Big Data

1. Descripción del caso:

Nuestra empresa, una Universidad Privada adscrita a la UMU, para establecer su nuevo plan estratégico con un horizonte temporal a 5-10 años, necesita estimar la demanda de alumnos de nueva entrada del grado de educación primaria.

Este proyecto de machine learning se centrará en predecir la demanda de alumnos de nuevo ingreso en el grado de educación primaria, utilizando datos demográficos y datos históricos de matriculación.

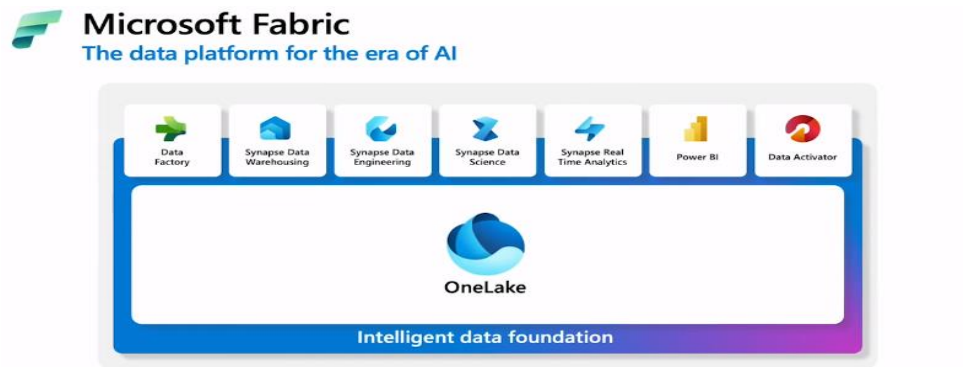
Para abordar el presente proyecto utilizaremos ***Microsoft Fabric***, una solución end-to-end para data science de reciente aparición en el mercado que abarca y facilita todo el proceso. Tanto la herramienta como el proyecto han sido seleccionados para mostar a la dirección de la empresa la importancia y el valor añadido que supondría instaurar este servicio de Microsoft en la empresa como ampliación a las herremientas de PowerPlatform que ya se tienen en uso. Este proyecto nos va a servir como primera aproximación al análisis de datos, de fácil y rápido despliegue, sobre un modelo de datos pequeño en un entorno embebido en nuestra propia plataforma de PowerBI mediante Microsoft Fabric.

En primer lugar definimos los distintos pasos:

1. Selección y definición del entorno y arquitectura que da soporte al proyecto.
2. Importación e ingesta de datos en nuestro Lakehouse
3. Análisis exploratorio de los datos
4. Entrenamiento y registro de un modelo de machine learning
5. Crear un reporte de PowerBI con las predicciones usando DirectLake

1. Microsoft Fabric

Plataforma de datos orientada a la ciencia de datos, donde de manera totalmente integrada podemos utilizar diferentes herramientas y lenguajes de programación para todo el ciclo de explotación del dato.



Primero crearemos dentro del servicio un entorno de trabajo para el proyecto llamado “ProyGH”, donde añadiremos un entorno virtual de almacenamiento de datos, denominado Lakehouse, al que nombraremos “PROYGH”, que nos servirá como soporte para la ingesta y preparación de los datos, reportes de BI, análisis de datos mediante el servicio de Synapse, y la inteligencia artificial y un servicio de análisis en tiempo real.

2. Ingeniería de Datos

Para nuestro objetivo del proyecto de obtener un modelo que nos permita predecir la demanda de matrículas de nuevo ingreso del Grado de Educación Primaria, tenemos las siguientes fuentes de datos:

- Datos de Matriculaciones.

Datos internos de nuestro centro sobre la oferta, demanda y matriculas del total de la universidad en la Región de Murcia, así como del total del grado en educación primaria tanto en nuestra universidad como en el resto. También datos de la nota de corte. También contaremos con los datos de gastos en la campaña de marketing para cada curso académico.

Estos datos sensibles de nuestro sistema de gestión están alojados en un servidor interno, y dado las características de los mismos hemos pedido que nos exporten un fichero .csv con los datos que podemos publicar.

El archivo es: DATOS_ISEN.csv

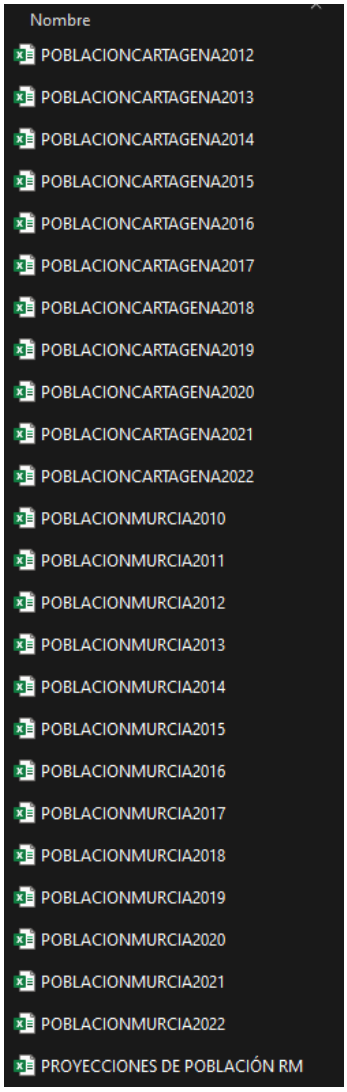
- Datos Demográficos.

Datos externos acerca del número de población censada por municipio y año, y la proyección a 14 años vista recogida del INE. Estos datos son de fuentes oficiales y están abiertos, aunque no tienen una API habilitada. Al ser datos no dinámicos, se ha optado por su descarga directa en formato .xlsx para su posterior manipulación, ya que por medio de webscraping nos hubiera tomado más tiempo y recursos.

El enlace a los datos es:

Cifras de población - CREM (carm.es)

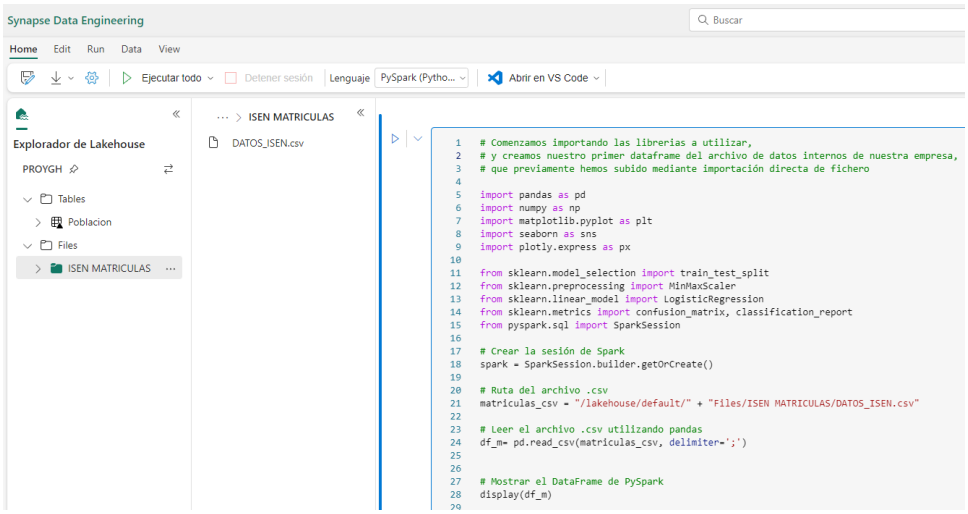
De aquí se han obtenido las siguientes tablas de población por edad y municipio, distinguiendo entre Cartagena y Región de Murcia



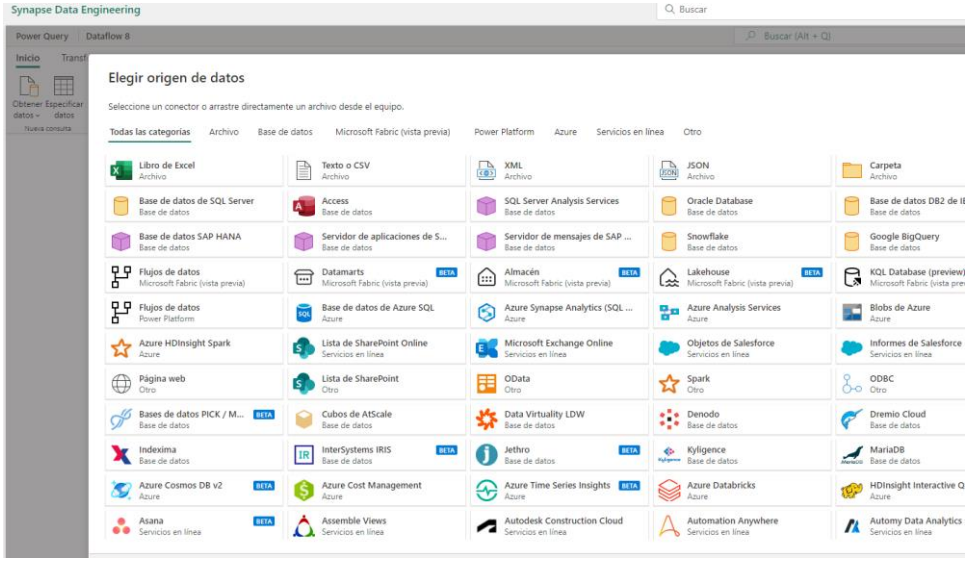
Una vez obtenidas las tablas con las que vamos a trabajar, abrimos nuestro Lakehouse para importar los datos.

En este punto para probar la versatilidad de MS Fabric, hemos utilizado dos formas distintas de obtención y depuración de los datos:

1) Con el archivo 'DATOS_ISEN.csv', utilizamos un Notebook integrado en el propio Lakehouse, seleccionando el lenguaje de PySpark(Python), para importar y cargar el dataframe. (*los detalles se muestran en el notebook adjunto).



2) Con los archivos de los datos de población utilizamos un Flujo de Datos Gen2, para traernos los datos ya ajustados manipulados en origen desde nuestro Sharepoint mediante el servicio de Power Query en lenguaje M:



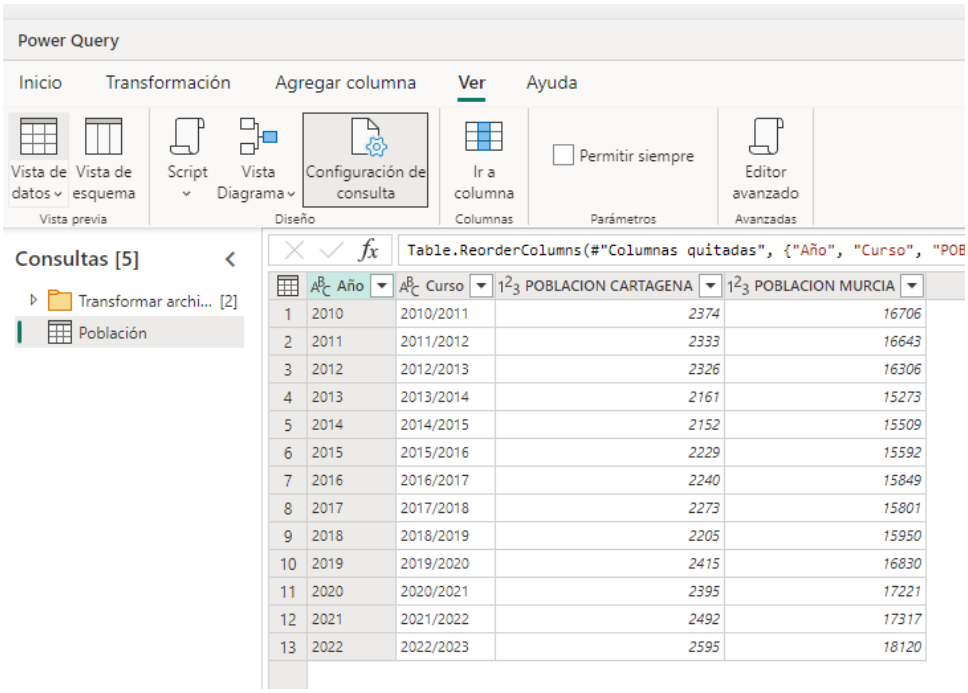
En este punto seleccionamos la carpeta de Sharepoint donde hemos alojado los archivos, y una vez traídos los transformamos mediante el siguiente script en M:

```
let

  Origen = SharePoint.Files("https://isenes.sharepoint.com/sites/Prcticas_ISEN", [ApiVersion = 15]),

  #"Filas filtradas" = Table.SelectRows(Origen, each Text.StartsWith([Name], "POBLACI")),

  #"Archivos ocultos filtrados" = Table.SelectRows(#"Filas filtradas", each [Attributes]?[Hidden]? <> true),

  #"Invocar función personalizada" = Table.AddColumn(#"Archivos ocultos filtrados", "Transformar archivo", each #"Transformar archivo"([Content])),

  #"Columnas con nombre cambiado" = Table.RenameColumns(#"Invocar función personalizada", {{"Name", "Source.Name"}}),
```

```
  #"Se han quitado otras columnas." = Table.SelectColumns(#"Columnas con nombre cambiado", {"Source.Name", "Transformar archivo"}),

  #"Columna de tabla expandida" = Table.ExpandTableColumn(#"Se han quitado otras columnas.", "Transformar archivo", Table.ColumnNames(#"Transformar archivo"(#"Archivo de ejemplo"))),

  #"Tipo de columna cambiado" = Table.TransformColumnTypes(#"Columna de tabla expandida", {{"Column1", type text}}),

  #"Texto recortado" = Table.TransformColumns(#"Tipo de columna cambiado", {{"Column1", each Text.Trim(_), type nullable text}}),

  #"Filas filtradas 1" = Table.SelectRows(#"Texto recortado", each Text.StartsWith([Column1], "De 18")),

  #"Personalizado agregado" = Table.AddColumn(#"Filas filtradas 1", "Población", each Text.Select([Source.Name],{"A".."Z"})),

  #"Personalizado agregado 1" = Table.AddColumn(#"Personalizado agregado", "Año", each Text.Select([Source.Name],{"0".."9"})),

  #"Se han quitado otras columnas. 1" = Table.SelectColumns(#"Personalizado agregado 1", {"Column2", "Población", "Año"}),

  #"Tipo de columna cambiado 1" = Table.TransformColumnTypes(#"Se han quitado otras columnas. 1", {{"Column2", Int64.Type}, {"Población", type text}, {"Año", type text}}),

  #"Columnas reordenadas" = Table.ReorderColumns(#"Tipo de columna cambiado 1", {"Población", "Año", "Column2"}),

  #"Columna condicional insertada" = Table.AddColumn(#"Columnas reordenadas", "Población Cartagena", each if Text.Contains([Población], "CARTAGENA") then [Column2] else 0),

  #"Columna condicional insertada 1" = Table.AddColumn(#"Columna condicional insertada", "Población Murcia", each if Text.Contains([Población], "MURCIA") then [Column2] else 0),

  #"Tipo de columna cambiado 2" = Table.TransformColumnTypes(#"Columna condicional insertada 1", {{"Población Cartagena", Int64.Type}, {"Población Murcia", Int64.Type}}),

  #"Filas agrupadas" = Table.Group(#"Tipo de columna cambiado 2", {"Año"}, {{"POBLACION CARTAGENA", each List.Sum([Población Cartagena]), type nullable number}, {"POBLACION MURCIA", each List.Sum([Población Murcia]), type nullable number}}),

  #"Tipo de columna cambiado 3" = Table.TransformColumnTypes(#"Filas agrupadas", {{"POBLACION CARTAGENA", Int64.Type}, {"POBLACION MURCIA", Int64.Type}}),

  #"Filas ordenadas" = Table.Sort(#"Tipo de columna cambiado 3", {{"Año", Order.Ascending}}),

  #"Tipo de columna cambiado 4" = Table.TransformColumnTypes(#"Filas ordenadas", {{"Año", Int64.Type}}),

  #"Personalizado agregado 2" = Table.AddColumn(#"Tipo de columna cambiado 4", "caño", each [Año] + 1),

  #"Tipo de columna cambiado 5" = Table.TransformColumnTypes(#"Personalizado agregado 2", {{"Año", type text}, {"caño", type text}}),

  #"Columna combinada insertada" = Table.AddColumn(#"Tipo de columna cambiado 5", "Curso", each Text.Combine({[Año], [caño]}, "/"), type text),

  #"Columnas quitadas" = Table.RemoveColumns(#"Columna combinada insertada", {"caño"}),
```

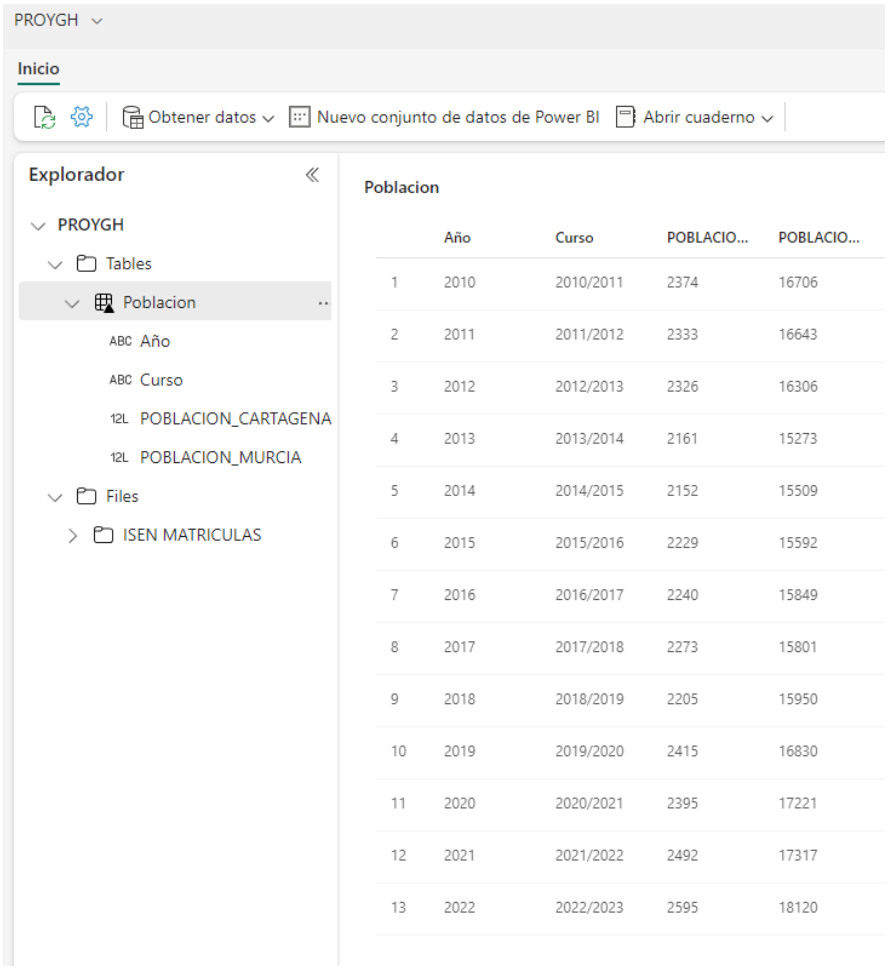
```
  #"Columnas reordenadas 1" = Table.ReorderColumns(#"Columnas quitadas", {"Año", "Curso", "POBLACION CARTAGENA", "POBLACION MURCIA"})

in

  #"Columnas reordenadas 1"
```

De esto nos resulta la siguiente tabla:



	Año	Curso	POBLACION CARTAGENA	POBLACION MURCIA
1	2010	2010/2011	2374	16706
2	2011	2011/2012	2333	16643
3	2012	2012/2013	2326	16306
4	2013	2013/2014	2161	15273
5	2014	2014/2015	2152	15509
6	2015	2015/2016	2229	15592
7	2016	2016/2017	2240	15849
8	2017	2017/2018	2273	15801
9	2018	2018/2019	2205	15950
10	2019	2019/2020	2415	16830
11	2020	2020/2021	2395	17221
12	2021	2021/2022	2492	17317
13	2022	2022/2023	2595	18120

Que el mismo servicio de Power Query nos lo almacena en nuestro Lakehouse como una tabla en formato Delta:



	Año	Curso	POBLACIO...	POBLACIO...
1	2010	2010/2011	2374	16706
2	2011	2011/2012	2333	16643
3	2012	2012/2013	2326	16306
4	2013	2013/2014	2161	15273
5	2014	2014/2015	2152	15509
6	2015	2015/2016	2229	15592
7	2016	2016/2017	2240	15849
8	2017	2017/2018	2273	15801
9	2018	2018/2019	2205	15950
10	2019	2019/2020	2415	16830
11	2020	2020/2021	2395	17221
12	2021	2021/2022	2492	17317
13	2022	2022/2023	2595	18120

Este flujo de datos se actualiza de forma automatizada, entregando una tabla Delta limpia para nuestro modelo.

3. Análisis exploratorio de los datos

Utilizamos el Notebook1 para terminar de limpiar los datos y mostrar diferentes medidas estadísticas (todo viene explicado en el notebook) utilizando PySpark.

Como resumen podemos comprobar en una primera aproximación que, el planteamiento original de que se podría predecir la demanda de nuevas matriculaciones en función del volumen de población no es realista, ya que vemos una correlación del 0.21 para la POBLACION_CARTAGENA y de -0.23 para POBLACION_MURCIA, que es de donde proceden el 98% de nuestros alumnos.

Otra observación relevante para el negocio es el comportamiento del GASTO_PUBLICIDAD que sorprendentemente tiene una correlación del –0.66, negativa con la demanda, por lo que se deberá replantear los medios de publicidad utilizados y su cuantía.

4. Entrenamiento y registro de un modelo de machine learning

Realizamos una regresión lineal múltiple seleccionando aquellas variables con una correlación mayor a la variable dependiente.

Los datos arrojados en el modelo de entrenamiento son:

R^2: 0.8296617805645987 Adjusted R^2: 0.7323256551729409 MAE: 5.453223426401498 MSE: 58.013175665072104 RMSE: 7.61663808153388

en vista de los resultados, el R2 nos da que el modelo está bien ajustado, sin embargo, el valor del MSE es muy alto para la variable a estimar, ya que los valores de la misma varian de 50 a 150, y el MSE es 58.01. Al no haber outliers, podemos pensar que el modelo este haciendo overfitting ()

Sin embargo los datos del modelo de Test:

R^2: 0.3722420164224697 Adjusted R^2: -2.1387899178876517 MAE: 12.013776283956778 MSE: 183.46227070053325 RMSE: 13.544824498698137

En este punto y como conclusión. después de haber hecho algunos ajustes entre variables, podemos concluir que el modelo no está generalizando bien los nuevos datos, necesita un ajuste o las variables y datos proporcionados no son lo suficientemente significativas para explicar la demanda.

5. Crear un reporte de PowerBI con las predicciones usando DirectLake

Adjunto .pbix con un breve reporte gráfico