

# **Sistema de recomendaciones de clases Life Tec de Monterrey**

(Proyecto de robótica)

**Sábado 28 de mayo del 2022**

## **Datos del estudiante:**

- **A01411787**
- **José de Jesús Bernal Mercado**
- Ing. en Sistemas Digitales y Robótica
- 9no semestre

## **Datos de los profesores:**

- **Ing. Sergio Camacho**
- **sergio.camacho@tec.mx**
- **Departamento de Ciencia de datos del Tec de Monterrey Campus Monterrey**
  - **Ing. Luz Eunice Angeles Ochoa**
  - **luz.eunice@tec.mx**
  - **Manuel Terán Melgarejo**
  - **teran@tec.mx**



# **TECNOLOGICO DE MONTERREY**

## Contenido

Investigación	3
Requerimientos	4
Descripción Técnica	5
Lista de herramientas utilizadas	6
Implementación paso a paso	6
Configuración del espacio de trabajo	6
Configuración de las herramientas a utilizar (Datasets y notebook)	8
Propiedades de los datasets	9
Desarrollo del modelo recomendador (SVD)	10
Entrenamiento del modelo	11
Deployment del modelo	11
Testing del modelo	13
Código de ejemplo	14
Métricas	15
Complejidad del problema	15
Mi experiencia	15
Conclusiones técnicas sobre el modelo	16
Referencias	16

## Definición del problema

Los sistemas de recomendación han dado de qué hablar en los últimos años debido a los avances tecnológicos que han surgido y su aplicación en las actividades de la vida diaria. Las grandes empresas han decidido dar un servicio personalizado a cada uno de sus clientes con el propósito de seguir incrementando sus ventas y dando una experiencia más certera al usuario. Uno de los ejemplos más conocidos es la plataforma de streaming Netflix, que con base en los gustos de un usuario hace recomendaciones de películas. Otro de los ejemplos más famosos es la tienda en línea de Amazon, que recomienda productos similares a los que los usuarios suelen comprar o ver. Sin duda los sistemas de recomendaciones son algo que evolucionarán de manera exponencial debido a la alta demanda que se ha generado en las diversas plataformas existentes, es por eso que para el Tecnológico de Monterrey es muy importante brindar un servicio personalizado en cuanto a la recomendación de cursos Life a los alumnos, con el propósito de que puedan tener una visión más clara y certera acerca de las decisiones a tomar al momento de elegir sus materias extracurriculares para que puedan desarrollarse en los diferentes ámbitos artísticos y deportivos que nuestra institución ofrece.

## Investigación

El propósito de este proyecto es poder desarrollar un modelo de Machine Learning que sea capaz de hacer recomendaciones de cursos Life a los alumnos dentro de la institución. Los modelos de machine learning han ido evolucionando bastante rápido debido a las diversas aplicaciones que han surgido en estos tiempos. Los modelos de sistemas de recomendaciones más comunes son los siguientes:

- Collaborative Recommender system
- Content-based recommender system
- Demographic based recommender system
- Utility based recommender system
- Knowledge based recommender system
- Hybrid recommender system

Nuestro problema cae en el caso de los Collaborative Recommender System, implementando específicamente un algoritmo de SVD (Singular Value Decomposition). SVD es comúnmente utilizado para hacer reducción de dimensiones, pero también se ha vuelto popular en los sistemas de recomendaciones. SVD está basado en un método de Matrix Factorization.

Matrix Factorization, como su nombre lo indica, factoriza una matriz en un producto de dos matrices. En la siguiente imagen podemos observar de manera gráfica la representación de la matriz  $X$  de tamaño  $(n \times p)$ , y el producto de las matrices  $A$  y  $B$ , de tamaño  $(n \times k)$  y  $(k \times p)$  respectivamente, siendo  $k$  el número de dimensiones latentes.

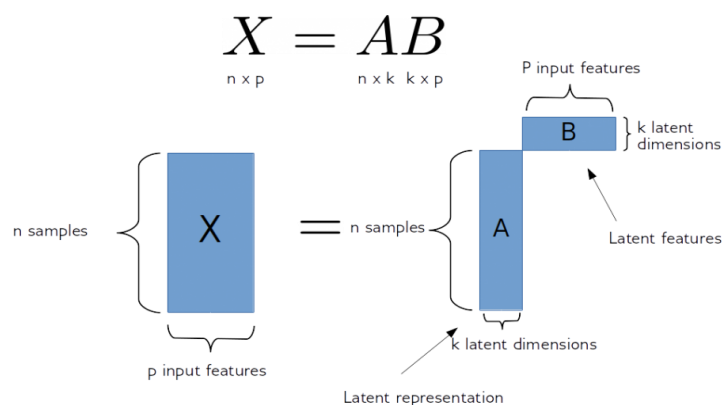
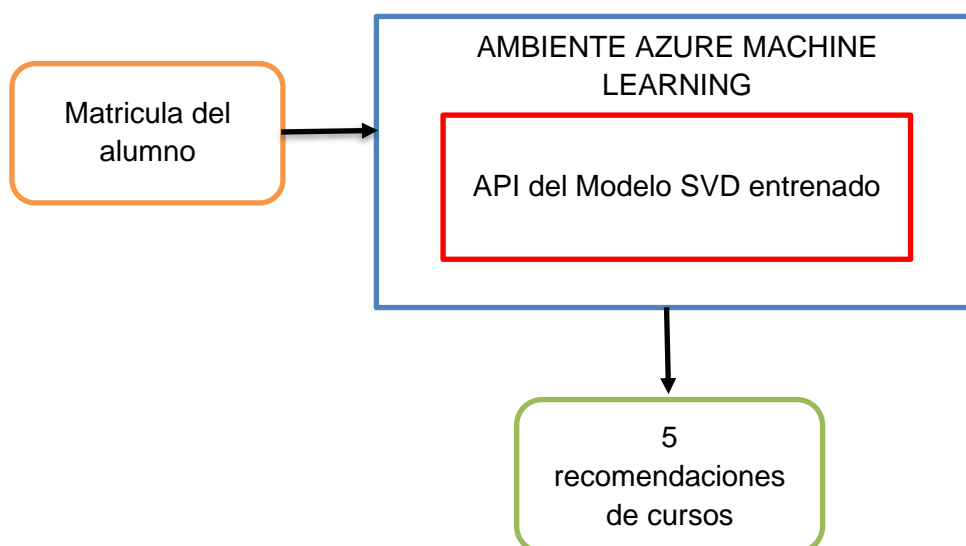


Ilustración 1 Descomposición de Matrices SVD

Para nuestro propósito, la matriz  $X$  será nuestra relación de alumnos y cursos Life que ellos ya han tomado, siendo los alumnos las filas, los cursos Life las columnas, y los valores los ratings dados en la encuesta ECOA, que la institución presenta cada semestre para evaluar las materias y profesores en una escala del 1 al 10. El número de dimensiones latentes nos ayuda a que nuestro modelo sea mejor, aunque si aumentamos demasiado el tamaño podemos empeorar nuestro modelo. (Analytics India, s.f.)

## Requerimientos



## Sample requirements form

name	Sistema de recomendaciones de cursos Life
purpose	Hacer recomendaciones de cursos life a los alumnos del tecnológico de monterrey con el propósito de que tengan una mejor experiencia al momento de elegir sus actividades extracurriculares
inputs	Matrícula del alumno
outputs	Lista de 5 recomendaciones de cursos Life
functions	Creación de dataset, Modelo SVD en la interfaz de Azureml
Metrics	MAE: 0.44, R2: 0.946, RMSE: 0.6, Explained Variance: 0946
manufacturing cost	100 credits

## Descripción Técnica

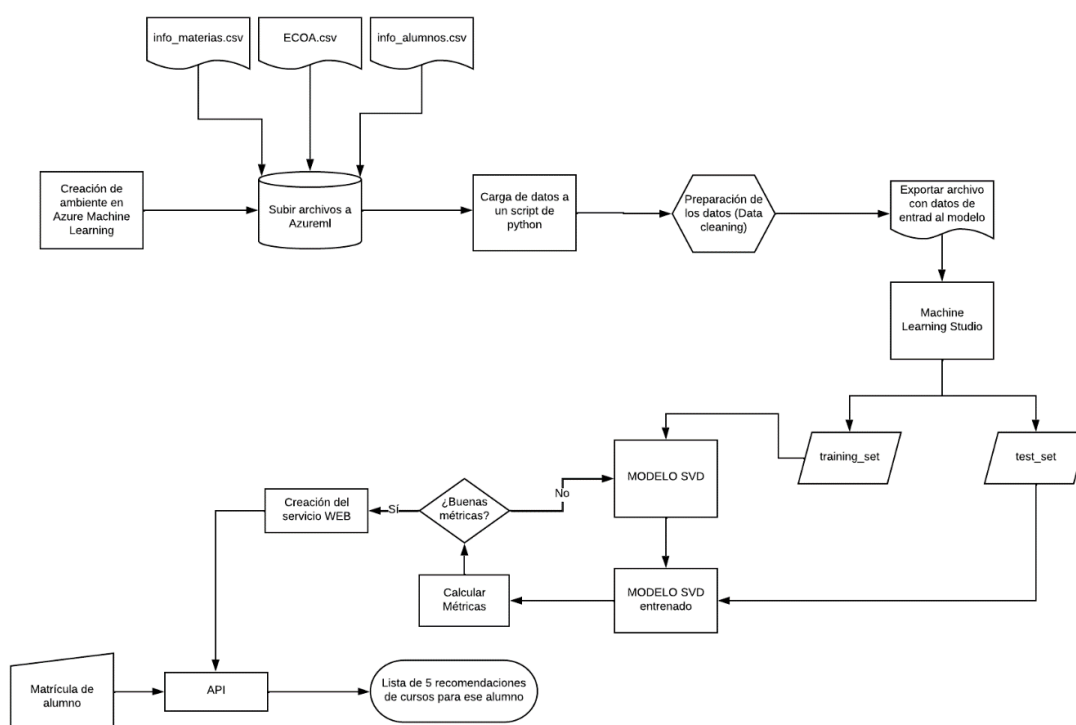


Ilustración 2 Diagrama Técnico del modelo

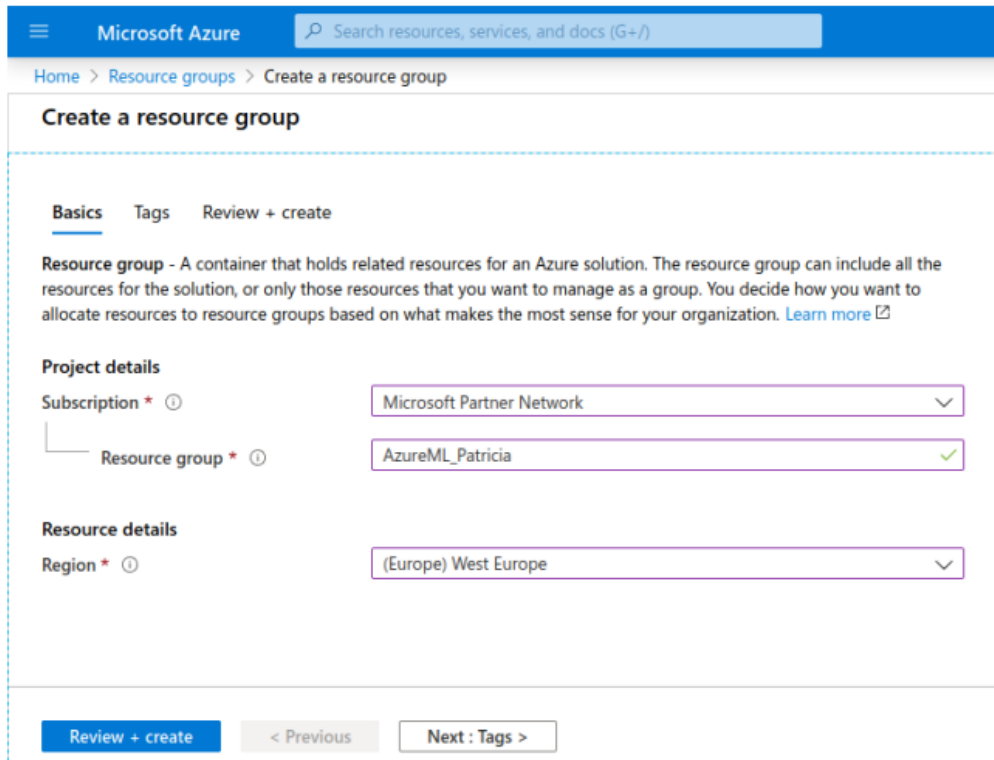
## Lista de herramientas utilizadas

- Python
- Pandas
- Numpy
- Scikit-Learn
- Azure Machine Learning Studio
- Jupyter Notebook
- VS Code

## Implementación paso a paso

### Configuración del espacio de trabajo

1.- Se creó un Grupo de recursos (Resource Group)



Microsoft Azure

Search resources, services, and docs (G+/)

Home > Resource groups > Create a resource group

### Create a resource group

**Basics** Tags Review + create

**Resource group** - A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. [Learn more](#)

**Project details**

Subscription \* ⓘ Microsoft Partner Network

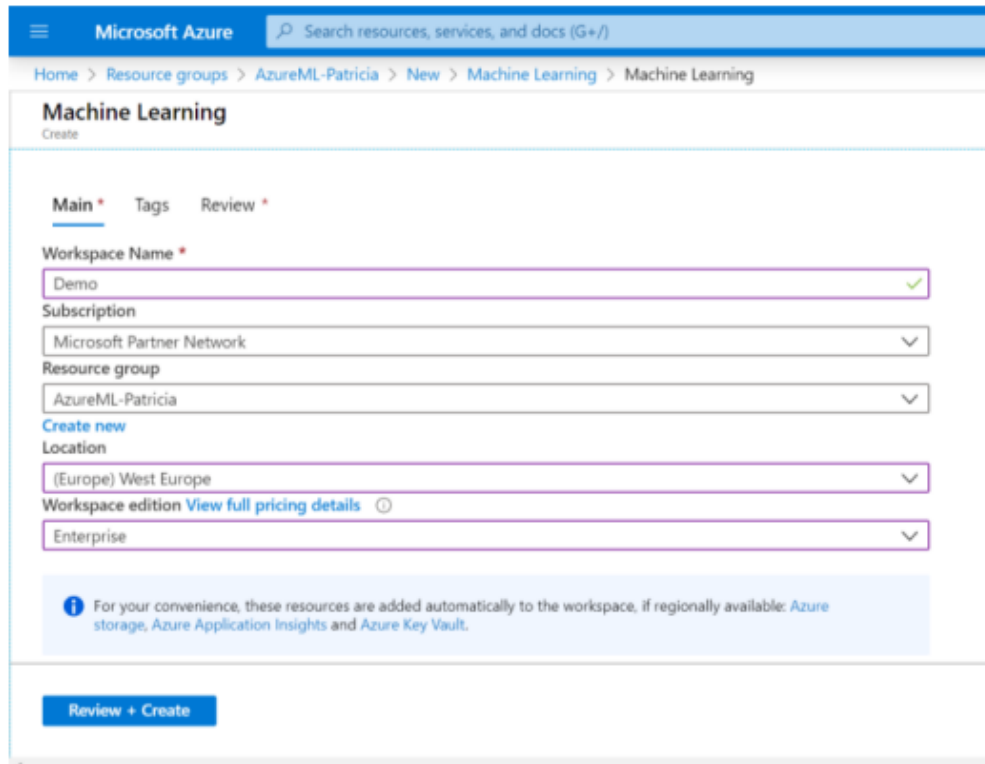
Resource group \* ⓘ AzureML\_Patricia

**Resource details**

Region \* ⓘ (Europe) West Europe

[Review + create](#) < Previous Next: Tags >

2.- Se creó un espacio de trabajo (Workspace)



Microsoft Azure Search resources, services, and docs (G+)

Home > Resource groups > AzureML-Patricia > New > Machine Learning > Machine Learning

### Machine Learning

Create

Main \* Tags Review \*

Workspace Name \* Demo ✓

Subscription Microsoft Partner Network ✓

Resource group AzureML-Patricia ✓

Create new

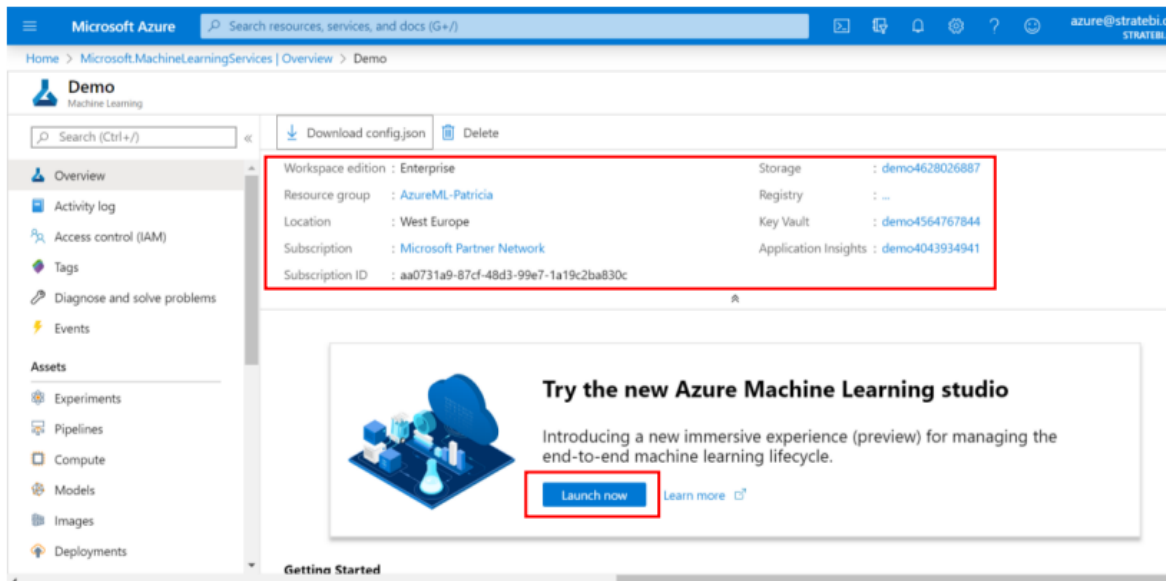
Location (Europe) West Europe ✓

Workspace edition View full pricing details ⓘ Enterprise ✓

For your convenience, these resources are added automatically to the workspace, if regionally available: Azure storage, Azure Application Insights and Azure Key Vault.

Review + Create

3.- Una vez creado el Workspace, se procedió a abrir Azure Machine Learning Studio



Microsoft Azure Search resources, services, and docs (G+)

Home > Microsoft.MachineLearningServices | Overview > Demo

**Demo**  
Machine Learning

Search (Ctrl+/) Download config.json Delete

Overview

- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Events

Assets

- Experiments
- Pipelines
- Compute
- Models
- Images
- Deployments

Workspace edition	: Enterprise	Storage	: demo4628026887
Resource group	: AzureML-Patricia	Registry	: --
Location	: West Europe	Key Vault	: demo4564767844
Subscription	: Microsoft Partner Network	Application Insights	: demo4043934941
Subscription ID	: aa0731a9-87cf-48d3-99e7-1a19c2ba830c		

**Try the new Azure Machine Learning studio**

Introducing a new immersive experience (preview) for managing the end-to-end machine learning lifecycle.

Launch now Learn more ⓘ

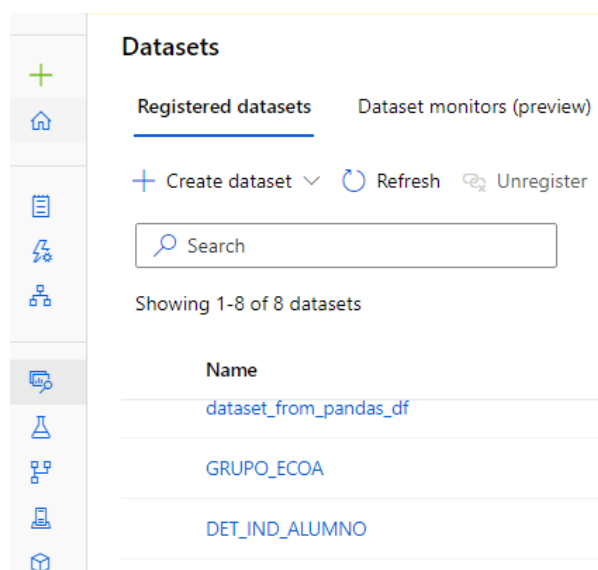
Getting Started

## Configuración de las herramientas a utilizar (Datasets y notebook)

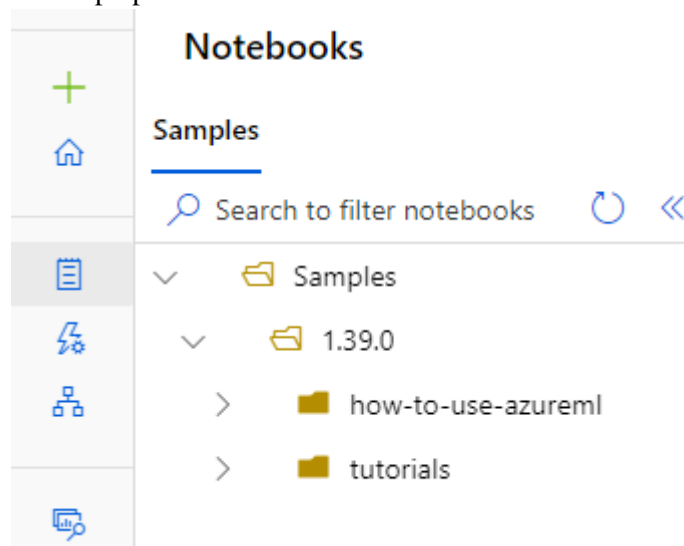
4.- Se agregaron los datasets a utilizar en la parte de Datasets de Azure Machine Learning Studio:

- GRUPO\_ECOA
- DET\_IND\_ALUMNO
- DWH\_MATERIAS\_EXTRA\_ACADEMICAS
- Enrollment\_fact
- Course\_dim
- User\_dim
- Enrollment\_term\_dim
- Pseudonym\_dim

NOTA\*: Todos las columnas ID's deben ser tipo **String**



5.- Se agregó el notebook de preparación de datos a la sección de Notebooks





## Propiedades de los datasets

### Dimensiones

- Enrollment\_fact : (4627975, 8)
- Course\_dim : (207089, 16)
- User\_dim : (272704, 17)
- Enrollment\_term\_dim : (41,7)
- Pseudonym\_dim : (276272, 19)
- GRUPO\_ECOA : (46572, 12)
- DET\_IND\_ALUMNO : (465921, 16)
- DWH\_MATERIAS\_EXTRA\_ACADEMICAS : (12974, 38)

En el notebook limpieza.ipynb se hace un merge y filtrado de todos estos datasets, teniendo un dataframe final con un tamaño de 83169 filas y 6 columnas.

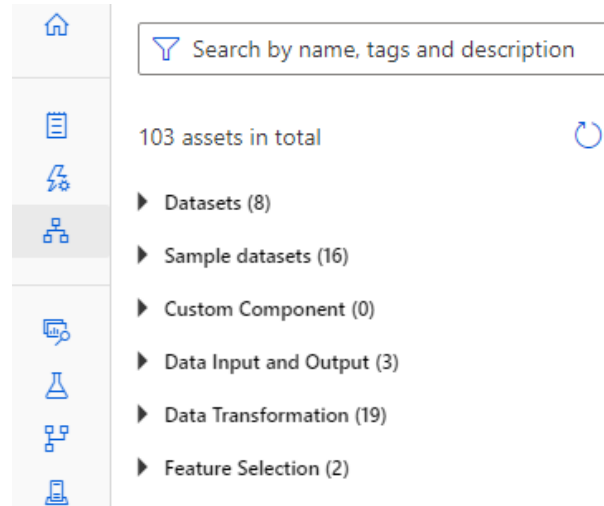
	matricula	user_name	code	NOMBRE MATERIA CORTO	CLAVE EJERCICIO ACADEMICO	ratings
0	A01411		XAFG3001	Acond físico en gimnasio	202011	8.0
1	A01411		XTOC4001	Sel tocho bandera fem mayor	202011	8.0
2	A01411		XAFG3002	Acond físico general	202113	5.0
3	A01411		XTOC4002	Sel tocho bandera var mayor	202113	4.0
4	A01411		KLID3002	Grupos estudiantiles	202011	10.0
5	A01411		XTOC4001	Sel tocho bandera fem mayor	202011	8.0
6	A01411		XTOC4001	Sel tocho bandera fem mayor	202013	6.0
7	A01411		KLID3002	Grupos estudiantiles	202011	10.0
8	A01411		YDCU3015	Danza urbana	202011	5.0
9	A01411		KLID3002	Grupos estudiantiles	202013	7.0
10	A01411		XTOC4001	Sel tocho bandera fem mayor	202013	8.0
11	A01411		KLID3001	Campamento de liderazgo	202111	7.0
12	A01411		KLID5001	Gobierno estudiantil	202011	6.0
13	A01411		KLID3002	Grupos estudiantiles	202013	6.0
14	A01411		XTOC4001	Sel tocho bandera fem mayor	202013	8.0
15	A01411		KLID3001	Campamento de liderazgo	202111	7.0
16	A01411		KLID5001	Gobierno estudiantil	202011	6.0
17	A01411		YCAN4002	Compañía canto	202011	8.0
18	A01411		KLID3002	Grupos estudiantiles	202013	8.0
19	A01411		XTOC4001	Sel tocho bandera fem mayor	202013	6.0

*Ilustración 3 DataFrame Final*

Los nombres de los estudiantes y matrículas fueron removidos de la imagen anterior por motivos de confidencialidad.

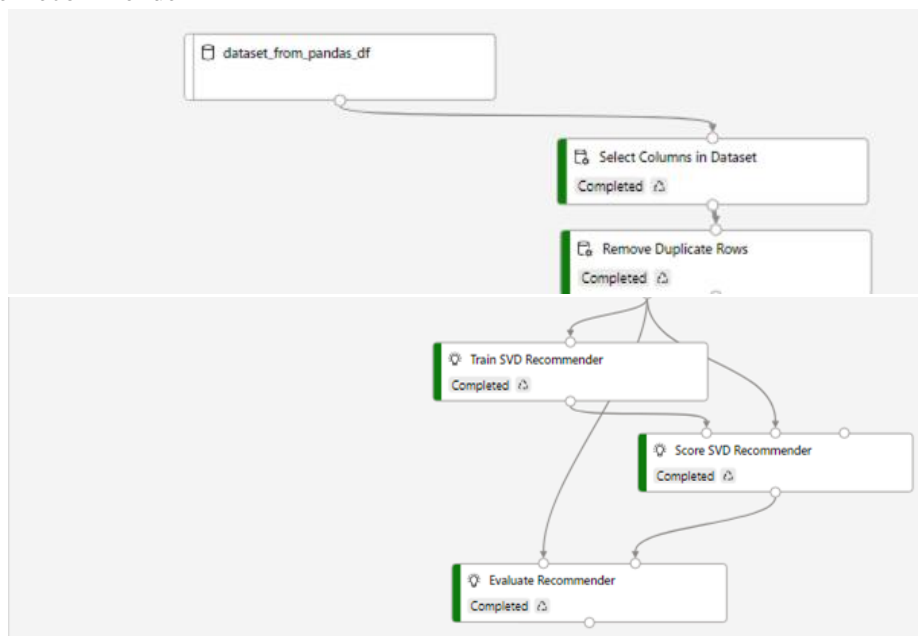
## Desarrollo del modelo recomendador (SVD)

6.- Se procedió a abrir la herramienta Designer, que contiene elementos que podemos usar para desarrollar nuestro modelo. (Microsoft Azure, s.f.)



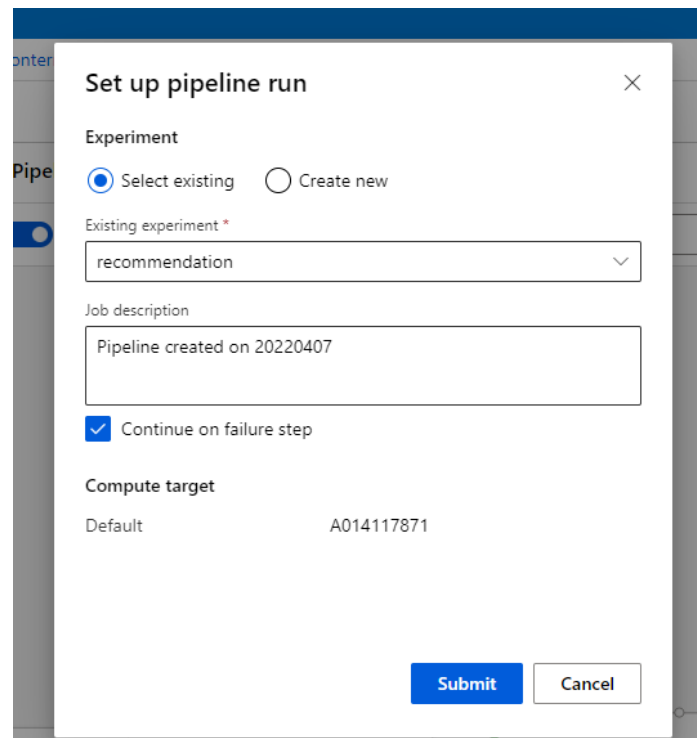
7.- Se agregaron los elementos necesarios para el Pipeline

- Dataset de salida del notebook de cleaning
- Select Columns Tool
- Remove Duplicate Rows
- Train SVD Recommender
- Score SVD Recommender
- Evaluate Recommender



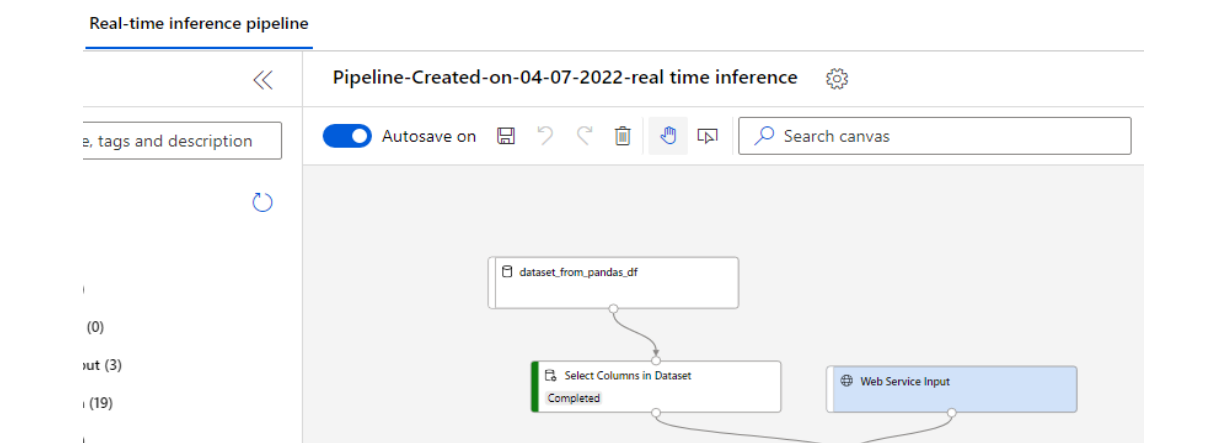
## Entrenamiento del modelo

8.- Para correr el pipeline se creó un Experiment. Una ventana como la de la imagen se abre al momento de dar submit. Una vez que se creó el Experiment, se corrió y entrenó el modelo con los datos dados. (Microsoft Azure, s.f.)

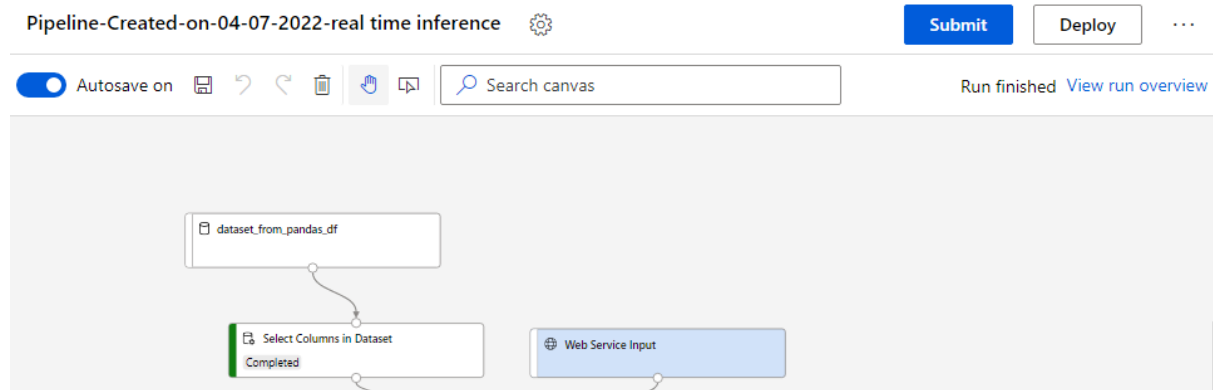


## Deployment del modelo

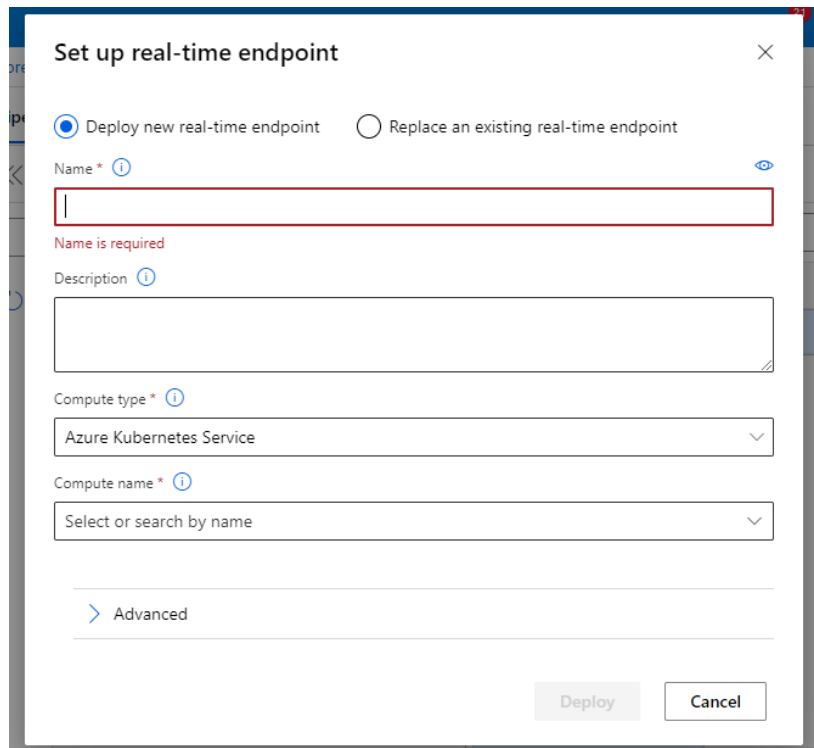
9.- Cuando el modelo terminó de entrenarse, se hizo el cambio Real Time Inference Pipeline para realizar la versión del pipeline en deployment. En esta parte automaticamente se agregan los cuadros de Web Service que Azure ya nos proporciona.



10.- Se corrió ese nuevo pipeline dando clic en Submit y posteriormente se dio clic en Deploy.

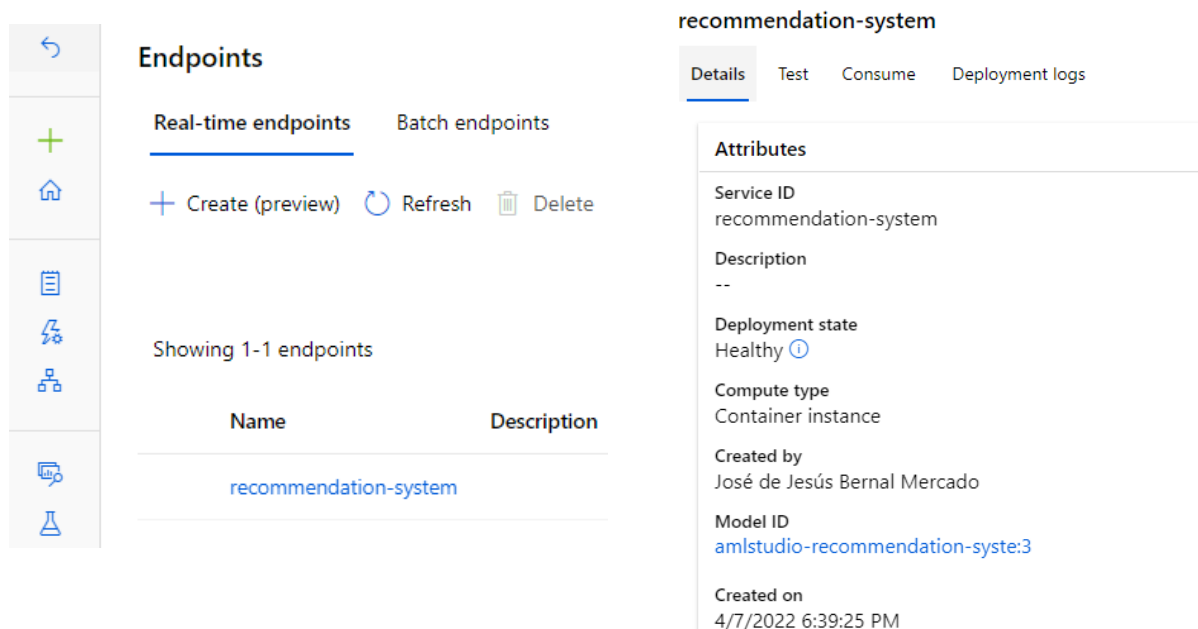


11.- Al dar clic en Deploy, se debe crear un Endpoint (Podemos elegir alguno que ya esté hecho o uno nuevo, en nuestro caso se creó uno nuevo). Al completar los campos se da clic en Deploy y el proceso empezará automáticamente.



## Testing del modelo

12.- Una vez terminado el proceso de deployment se podrá observar el estado del Endpoint en la pestaña de endpoints de AzureML. Al dar clic en el nombre del endpoint se podrán observar los detalles del deployment.



The screenshot shows the AzureML 'Endpoints' page. On the left, there's a sidebar with navigation icons. The main area is titled 'Endpoints' and has tabs for 'Real-time endpoints' and 'Batch endpoints'. Below these, there are buttons for '+ Create (preview)', 'Refresh', and 'Delete'. A message says 'Showing 1-1 endpoints'. A table lists the endpoint 'recommendation-system'. On the right, the 'recommendation-system' details are shown, with tabs for 'Details', 'Test', 'Consume', and 'Deployment logs'. The 'Details' tab is active, showing attributes like Service ID, Description, Deployment state (Healthy), Compute type (Container instance), Created by (José de Jesús Bernal Mercado), Model ID (amlstudio-recommendation-syste:3), and Created on (4/7/2022 6:39:25 PM).

En la pestaña de Consume se puede observar el código necesario para hacer requests al modelo. De igual forma también se proporciona el url y la apikey.



The screenshot shows the 'recommendation-system' endpoint details, specifically the 'Consume' tab. It displays a code editor with Python code for making a request to the endpoint. The code includes the following lines:

```

15     "inputs": {
16     },
17     "GlobalParameters": {
18     }
19 }
20
21 body = str.encode(json.dumps(data))
22
23 url = 'http://66fb1a8b-6f16-401a-898f-214624fd3ff5.westeurope.azurecontainer.io/score'
24 api_key = 'F4zqURVESRN0OrjvNknHwVJmfrEo1Ye8' # Replace this with the API key for the web
25 headers = {'Content-Type': 'application/json', 'Authorization': ('Bearer ' + api_key)}
26
27 req = urllib.request.Request(url, body, headers)
28

```

En la pestaña de Test se puede observar como debe ser el input al momento de hacer el request y también se puede observar la respuesta que entrega.

recommendation-system

Details **Test** Consume Deployment logs

Input data to test real-time endpoint **Test** Test result

```

{
  "Inputs": {
    "WebServiceInput0": [
      {
        "matricula": "A00226905"
      },
      {
        "matricula": "A00226905"
      },
      {
        "matricula": "A00227214"
      }
    ]
  },
  "GlobalParameters": {}
}

```

```

{
  "Results": {
    "WebServiceOutput0": [
      {
        "User": "A00226905",
        "Recommended Item 1": "XBA55002",
        "Recommended Item 2": "KWEL3002",
        "Recommended Item 3": "XTOC5003",
        "Recommended Item 4": "XTOC5002",
        "Recommended Item 5": "XAFG3003"
      },
      {
        "User": "A00227214",
        "Recommended Item 1": "KWEL3002",
        "Recommended Item 2": "XTOC5002",
        "Recommended Item 3": "VVDI3001",
        "Recommended Item 4": "XTOC5003",
        "Recommended Item 5": "XAFG3003"
      }
    ]
  }
}

```

## Código de ejemplo

```

import json
import urllib

input_data = {'matricula' : 'A01411787',
              'matricula' : 'A01567634',
              'matricula' : 'A09768241'}

data = {
  'Inputs' : {'WebServiceInput0':[input_data]},
  'GlobalParameters':{}}
}

body = str.encode(json.dumps(data))

url = 'http://.....'
api_key = '.....'
headers = {'Content-Type' : 'application/json', 'Authorization':('Bearer' + api_key)}

req = urllib.request.Request(url, body, headers)
response = urllib.request.urlopen(req)

result = response.read()
print(result)

```

Ilustración 5 Código para hacer un request a la API del modelo

## Métricas

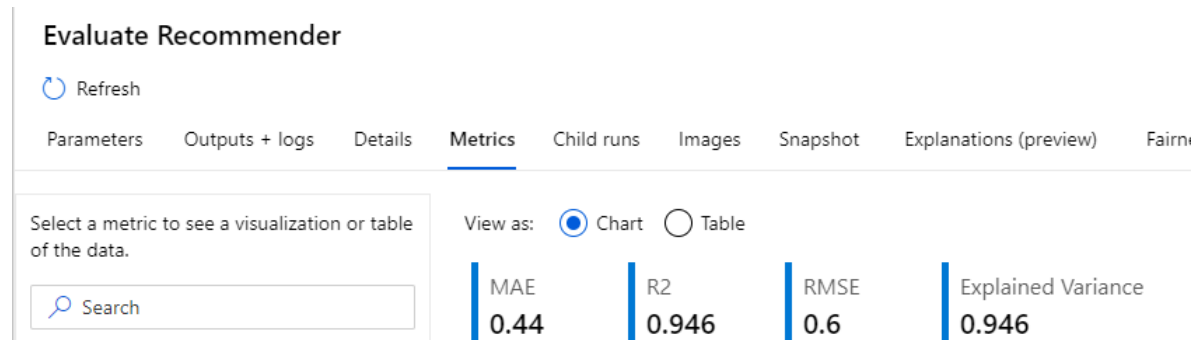
Las métricas que se usaron para la evaluación del recomendador fueron las siguientes:

MAE: 0.44,

R2: 0.946

RMSE: 0.6

Explained Variance: 0.946



(Microsoft Azure, s.f.)

## Complejidad del problema

Creo que una de las ventajas de trabajar en un proyecto de ciencia de datos en la actualidad es que tenemos al alcance un volumen de datos bastante grande para lo que requerimos. Pero a la vez una de las desventajas es que hay que saber elegir bien los datos con los que vamos a trabajar para poder tener resultados eficientes al terminar el proyecto. En el caso de este recomendador tuvimos la fortuna de contar con un equipo de ingeniería de datos que nos brindó los datos de forma limpia facilitándonos esa parte del proceso, y permitiendo que nos enfocáramos en la parte solamente del modelo predictivo.

## Mi experiencia

Este es mi primer proyecto en el que hago un recomendador de ítems, sin duda fue de gran aprendizaje en muchos aspectos, tanto en la parte matemática al entender como trabaja el SVD y en la parte de sistemas aprendiendo a usar Azure Machine Learning, que nos brindó herramientas poderosas y de fácil uso. Estoy muy contento con mi participación en este proyecto y espero que se pueda tomar como base para mejorarlo cada vez que sea posible.

## Conclusiones técnicas sobre el modelo

Podemos concluir que nuestro modelo entrenó de manera eficiente los datos dándonos una  $R^2$  del 94% en el set de testing, evitando el overfitting y malas recomendaciones. Es un modelo simple al que realmente le podemos agregar más cosas, tales como características de cada clase o de cada alumno, pero por ahora es un modelo funcional bastante robusto.

## Referencias

- Analytics India*. (s.f.). Obtenido de • <https://analyticsindiamag.com/singular-value-decomposition-svd-application-recommender-system/>
- Microsoft Azure*. (s.f.). Obtenido de <https://azure.microsoft.com/es-mx/blog/building-recommender-systems-with-azure-machine-learning-service/>
- Microsoft Azure*. (s.f.). Obtenido de <https://docs.microsoft.com/en-us/azure/machine-learning/component-reference/train-svd-recommender>
- Microsoft Azure*. (s.f.). Obtenido de <https://docs.microsoft.com/en-us/azure/machine-learning/component-reference/score-svd-recommender>