

# Informe Sistema de Recomendación

Lic. Simontacchi Lautaro

[lauta.logica@gmail.com](mailto:lauta.logica@gmail.com)

## Introducción:

Con el objetivo de confeccionar un sistema de recomendación de películas se realizó el siguiente proyecto donde se aplicaron herramientas de machine learning (ML), programación Python y estadística para la limpieza de datos, análisis y procesamiento de los mismos, y creación de una aplicación de consulta y recomendación de películas para ser utilizada mediante una interfaz de programación de aplicaciones (API) de manera online.

En el siguiente informe se presentan, de manera detallada, todos los pasos seguidos, desde la información de los datos originales hasta la implementación de la API final.

## Datos:

Se recibió dos archivos con los datos para ser utilizados, estos son movies\_dataset.csv y credits.csv.

Respecto a movies\_dataset, este archivo cuenta con 24 columnas con 45460 filas. Estas son:

Nombre:	Tipo de objeto
adult	object
belongs_to_collection	object
budget	object
genres	object
homepage	object
id	object
imdb_id	object
original_language	object
original_title	object
overview	object
popularity	object
poster_path	object
production_companies	object
production_countries	object
release_date	object
revenue	float64
runtime	float64
spoken_languages	object
status	object
tagline	object
title	object
video	object
vote_average	float64
vote_count	float64

Se presentan las primeras 5 filas del dataset para tener un mejor entendimiento del tipo de datos que contiene cada columna:

costo	id	lenguaje	popularidad	lanzamiento	ingresos	Duración	Estado	Título	Promedio Votos	Votos Totales	Saga	Compañía	Año Lanzamiento
30000000	862	en	21,946943	1995-10-30	373554033	81	Released	Toy Story	7,7	5415	Toy Story Collection	Pixar Animation Studios	1995
65000000	8844	en	17,015539	1995-12-15	262797249	104	Released	Jumanji	6,9	2413	NaN	TriStar Pictures	1995
0	15602	en	11,7129	1995-12-22	0	101	Released	Grumpier Old Men	6,5	92	Grumpy Old Men Collection	Warner Bros.	1995
16000000	31357	en	3,859495	1995-12-22	81452156	127	Released	Waiting to Exhale	6,1	34	NaN	Twentieth Century Fox Film Corporation	1995
0	11862	en	8,387519	1995-02-10	76578911	106	Released	Father of the Bride Part II	5,7	173	Father of the Bride Collection	Sandollar Productions	1995

Tabla 1: muestra de las columnas del dataset entregado "movies\_dataset.csv" con el que se obtuvo los datos para ser utilizados en la API

Las columnas restantes, que corresponde a objetos están anidadas, es decir corresponden a diccionarios o listas. Las mismas serán analizadas en el siguiente punto.

Mientras que credits.csv cuenta con 3 columnas de 45476 filas. Estas son.

cast object  
crew object  
id int64

Aunque estén anidadas observamos las filas.

cast	crew	id
[{'cast_id':	[{'credit_id':	862
[{'cast_id':	[{'credit_id':	8844
[{'cast_id':	[{'credit_id':	15602
[{'cast_id':	[{'credit_id':	31357
[{'cast_id':	[{'credit_id':	11862

Tabla 2: La tabla muestra las primeras filas del dataset credits.csv, el mismo tiene columnas anidadas que fueron procesadas para ser utilizadas en las consultas.

Ambas filas corresponden a diccionarios anidados.

### Pre Procesamiento:

Con el fin de reducir el tamaño de los archivos de datos se realizó una limpieza de los mismos eliminando filas duplicadas y columnas que no sean de interés para el análisis y/o consultas. También se eliminaron las columnas anidadas, habiéndolas desanidado previamente. Y se rellenaron los valores nulos con "0".

A continuación, una comparación entre el dataset antes y después de la limpieza:

Tamaño del dataset original: 45376 filas y 14 columnas

Tamaño del dataset sin duplicados: 45343 filas y 10 columnas

NULOS DATASET ORIGINAL		NULOS DATASET UTILIZADO	
budget	0	budget	0
id	0	id	0
original_language	11	popularity	0
popularity	0	release_date	0
release_date	0	revenue	0
revenue	0	runtime	246
runtime	246	status	80
status	80	title	0
title	0	vote_average	0
vote_average	0	vote_count	0
vote_count	0	release_year	0
belongs_to_collection_name	40888	return	0
production_companies_names	11796		
release_year	0		

Ilustración 1: La imagen de la izquierda muestra los nulos presentes en el dataset original. La imagen de la izquierda se observa que en el dataset utilizado se corrigieron estos nulos, ya sea eliminando la columna o rellenando la misma con ceros

Respecto a las columnas anidadas se crearon los siguientes archivos que fueron utilizados en el API:



directores\_api: contiene dos columnas “name”: con el nombre del actor e “id\_original”: para hacer el join con el dataset principal.

actores\_api: contiene dos columnas “name”: con el nombre del actor e “id\_original”: para hacer el join con el dataset principal.

genres\_api: contiene las columnas “id” con el ID de género de la película, “name”: con el género de la película e “id\_original”: para hacer el join con el dataset principal.

#### Análisis de los datos:

Primero se realizó una estadística básica de las variables numéricas

	Costo	popularidad	ingresos	voto promedio	conteo de votos	retorno
<b>Total</b>	45343	45343	45343	45343	45343	45343
<b>media</b>	4,24E+06	2,926469	1,12E+07	5,624306	110,14408	6,61E+02
<b>Desv Est.</b>	1,74E+07	6,011006	6,44E+07	1,915151	491,914238	7,47E+04
<b>min</b>	0,00E+00	0	0,00E+00	0	0	0,00E+00
<b>Percentil 25%</b>	0,00E+00	0,38891	0,00E+00	5	3	0,00E+00
<b>Percentil 50%</b>	0,00E+00	1,130302	0,00E+00	6	10	0,00E+00
<b>Percentil 75%</b>	0,00E+00	3,691946	0,00E+00	6,8	34	0,00E+00
<b>max</b>	3,80E+08	547,488298	2,79E+09	10	14075	1,24E+07

Tabla 3: En la tabla se muestran las estadísticas básicas de las columnas numéricas del dataset utilizado en el API

Lo más interesante que se obtiene de la tabla anterior es que en promedio (media) el costo por película es de 4240000 u\$s mientras que el promedio de ingresos es de

11200000 u\$s por lo que las ganancias son del 265 %. Mientras que el puntaje promedio de las reseñas es de 5,6. La columna retorno se obtuvo dividiendo las columnas ingresos/costos, con la aclaración de que cuando no hay datos disponibles para el cálculo se puso 0 como valor.

No se decidió quitar ningún dato que pueda considerarse outlier ya que los datos numéricos son propios de cada película, y no son el resultado de una medición por ejemplo, por lo que quitarlos sería quitar información importante.

Lo siguiente que se realizó es determinar si existe una correlación, es decir una relación del tipo lineal, entre los datos, se muestra a continuación la matriz de correlación y los gráficos de dispersión.

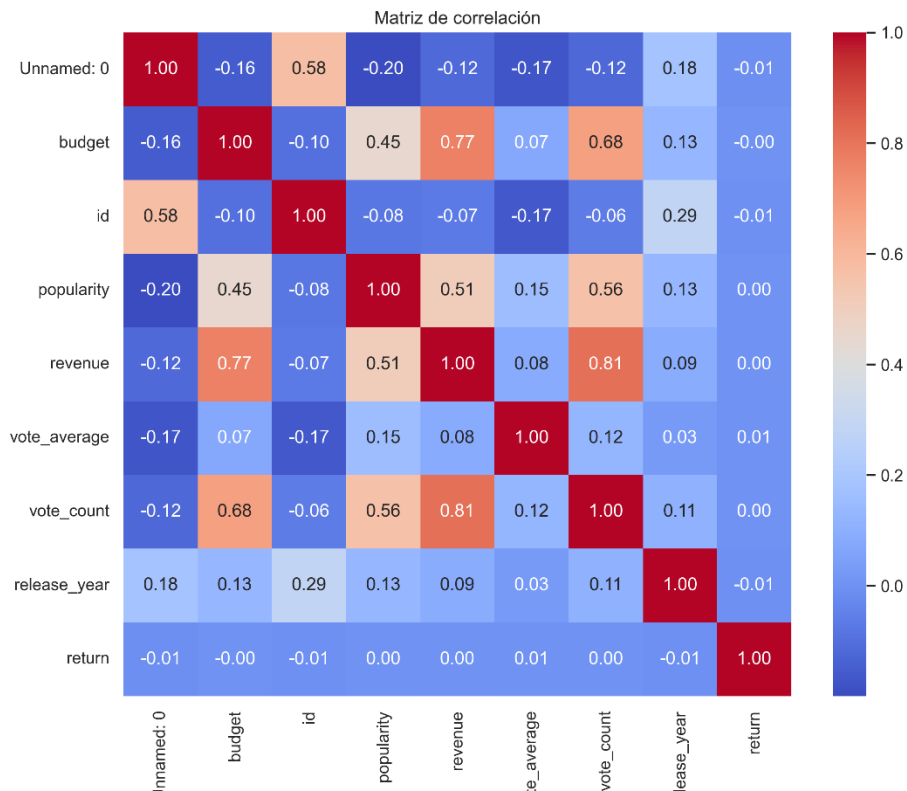
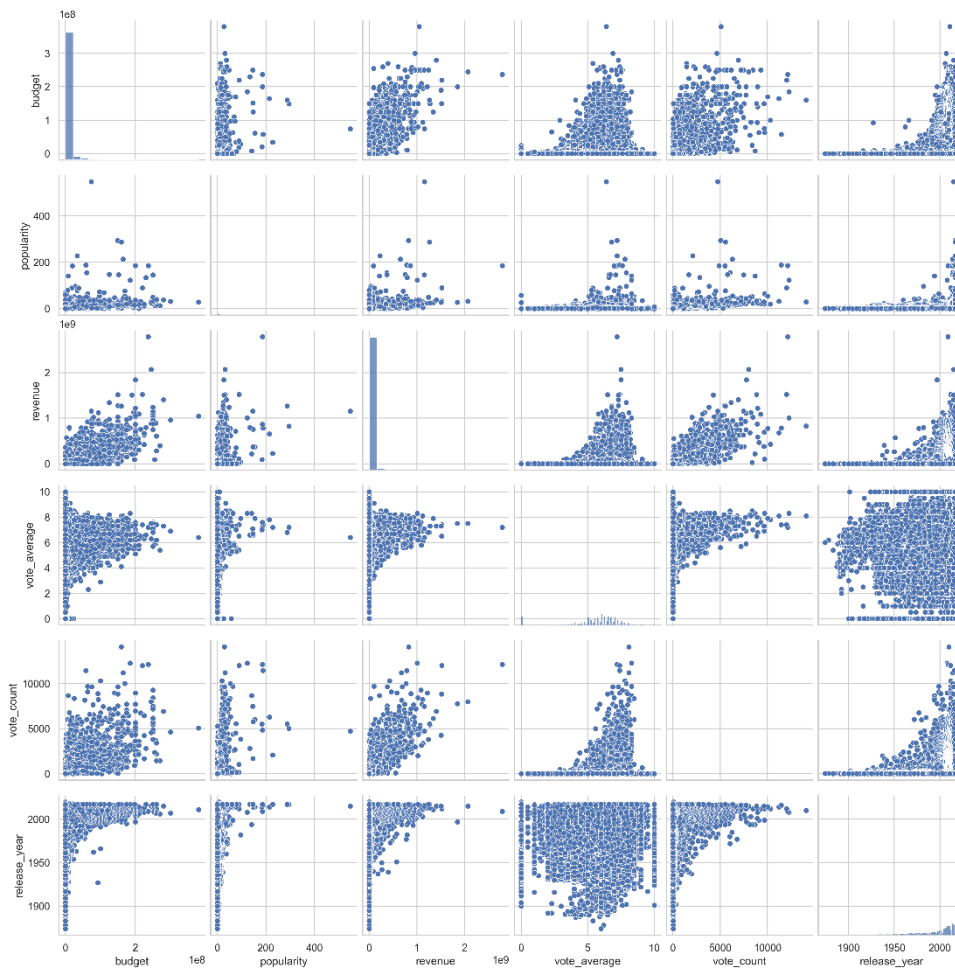


Ilustración 2: Matriz de correlación. La misma se utiliza para ver cuan relacionados se encuentran los datos. Siendo 1 los datos están correlacionados y 0 los datos son independientes entre si.



*Ilustración 3: gráficos de dispersión cruzada entre los datos numéricos. Estos gráficos se utilizan para determinar una relación/dependencia lineal entre los mismos.*

La matriz de dispersión nos muestra que las variables recaudación (revenue) con el número de votos recibido (vote\_count) tienen un coeficiente de correlación de 0,81 por lo que se puede asumir una relación lineal entre ambas. Esta se corrobora mirando el gráfico de dispersión ya que las películas con mejor puntaje tienen una mayor recaudación. Una correlación similar, pero menor, se da entre estas variables con el presupuesto (Budget) y la popularidad (popularity). Un dato interesante se observa en el gráfico de dispersión entre estas últimas variables, presupuesto con popularidad, ya que se observa que las películas con menor presupuesto tienen una popularidad similar a las de alto presupuesto, observándose tanto películas de bajo presupuesto como de alto con gran popularidad. Finalmente, como dato extra, si miramos en el gráfico de dispersión a la columna donde se encuentra la relación con el año de lanzamiento podemos observar que tanto presupuesto como ganancia van en aumento año tras año.

Ahora vemos algunas gráficas de frecuencia.

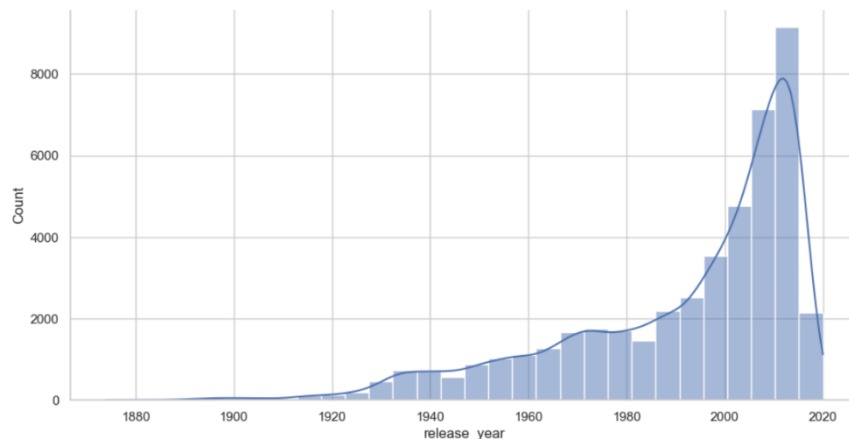


Ilustración 4: En el gráfico se representa la cantidad de películas por año de lanzamiento.

La gráfica 1 corrobora algo que es lógico ya que se ve que la cantidad de películas estrenadas por año va en aumento. Un dato extra es que el dataset no contiene completo al último año (2020) por lo que

es necesario una actualización del mismo.

Analizando las películas por género la ilustración 2 se observa la cantidad de películas por género.

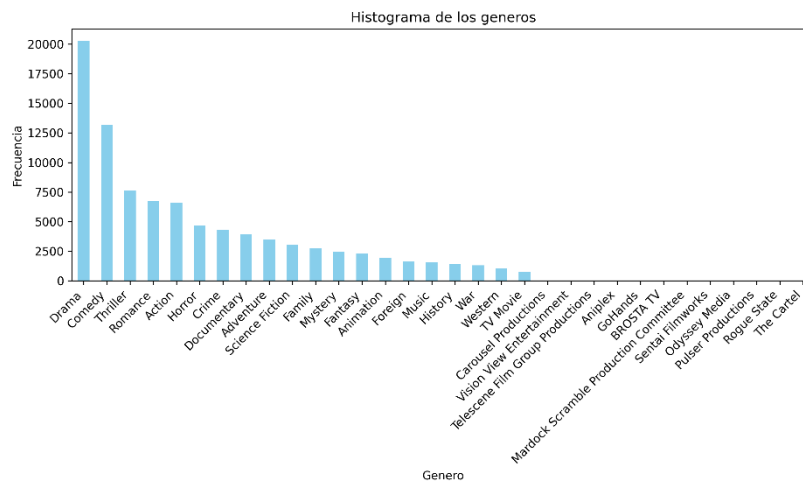


Ilustración 5: En el gráfico se muestra la frecuencia de películas según el género de las mismas. Cave aclarar que algunas películas están clasificadas con más de un género.

Las películas de Drama son las de mayor producción llegando a casi 20000 de este género, seguida por las comedias con un poco más de 12500 y luego los géneros más producidos son Thriller, las películas románticas y las de acción. Recordemos que algunas películas son caracterizadas

con más de un género. Por ejemplo, la película “Toy Story” es una comedia familiar de animación.

Para completar el análisis de los datos se realiza un análisis respecto a las compañías productoras, los directores y los actores.

Respecto a las compañías tenemos que existen 23537 compañías. Se muestran las siguientes tablas donde se observan las 10 compañías con mayores ganancias totales y las que mas películas realizó. Siendo Warner Bros la que lidera ambas categorías. Se observa también la gran ventaja que tienen los EEUU sobre el resto de los países, ya que ambas tablas se componen de empresas de ese país. La excepción es Canal + de origen francés. No se muestra la ganancia por película ya que muchas de las productoras solo realizaron una película y no tiene sentido realizar esta operación.

Ganancias totales		Películas Totales	
Warner Bros.	6,3525E+10	Warner Bros.	1250
Universal Pictures	5,5259E+10	Metro-Goldwyn-Mayer (MGM)	1080
Paramount Pictures	4,8886E+10	Paramount Pictures	1007
Twentieth Century Fox	4,7688E+10	Twentieth Century Fox	836
Walt Disney Pictures	4,0837E+10	Universal Pictures	830
Columbia Pictures	3,228E+10	Columbia Pictures Corporation	448
New Line Cinema	2,2173E+10	Canal+	442
Amblin Entertainment	1,7344E+10	Columbia Pictures	431
DreamWorks SKG	1,5476E+10	RKO Radio Pictures	290
Dune Entertainment	1,5004E+10	United Artists	279

Tabla 4: se muestran las 10 compañías que más ganancias obtuvieron (izquierda) y que más películas produjeron (derecha)

Haciendo un análisis similar con los directores obtenemos los siguientes resultados:

mayor ganancia		mayor cantidad	
Steven Spielberg	9256621422	John Ford	68
Peter Jackson	6528244659	Michael Curtiz	65
Michael Bay	6437466781	Werner Herzog	55
James Cameron	5900610310	Alfred Hitchcock	53
David Yates	5334563196	Georges Méliès	51
Christopher Nolan	4747408665	Jean-Luc Godard	50
Robert Zemeckis	4138233542	Woody Allen	49
Tim Burton	4032916124	Sidney Lumet	46
Ridley Scott	3917529240	Charlie Chaplin	44
Chris Columbus	3866836869	Henry Hathaway	43

Tabla 5: La tabla muestra los 10 directores que más ganancia obtuvieron (izquierda) y los 10 directores que más películas realizaron (derecha)

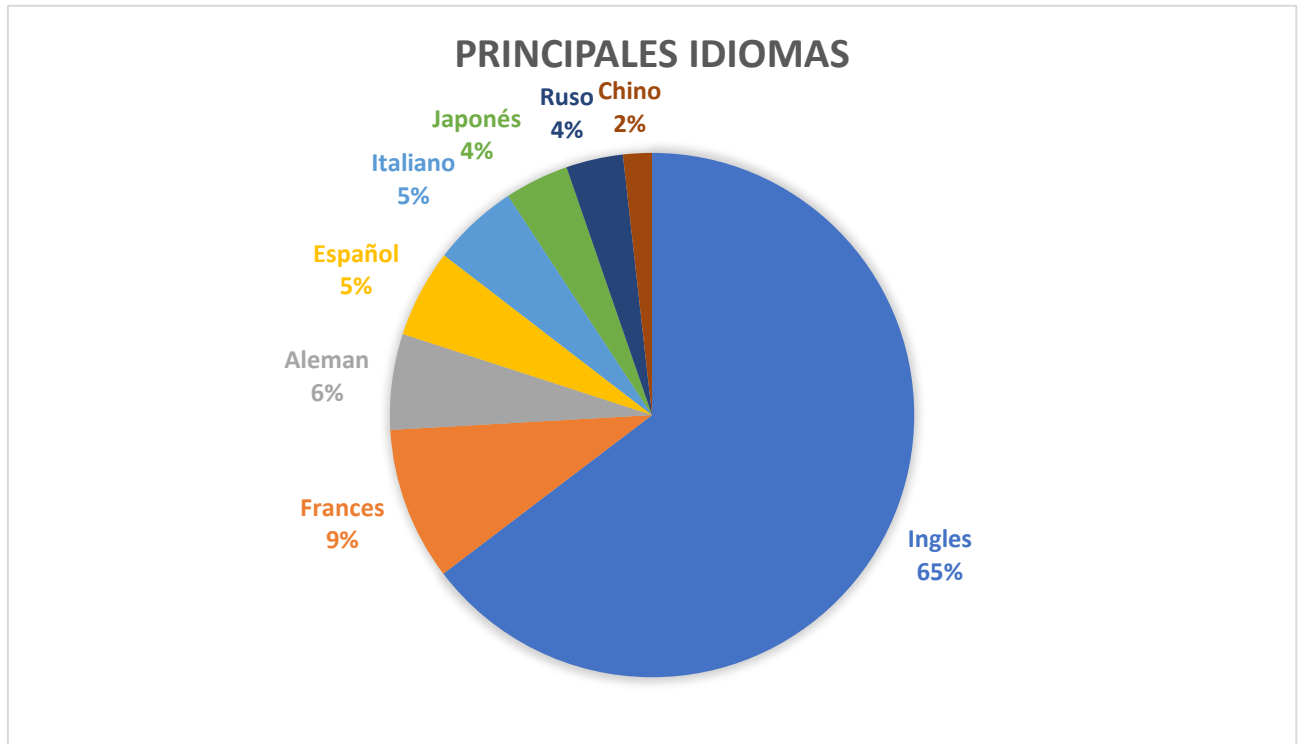
Los directores contemporáneos son los que mayor ganancia tienen por película, lo que es lógico ya que tanto los costos como las ganancias aumentaron con el tiempo. Siendo Steven Spielberg el que más retornos tiene.

Para corroborar la supremacía de los EEUU tenemos las siguientes tablas:

Ganancias totales		Películas Totales	
United States of America	4,7668E+11	United States of America	21179
United Kingdom	7,3433E+10	United Kingdom	4100
Germany	2,9175E+10	France	3956
France	2,0296E+10	Germany	2264
Canada	1,8504E+10	Italy	2175
Australia	1,167E+10	Canada	1765
China	1,1384E+10	Japan	1654
Japan	9492467119	Spain	964
New Zealand	8805470541	Russia	912
India	6265191760	India	830

Se observan que las películas de origen estadounidense tienen una ganancia que tienen un orden de magnitud más que las del segundo país en ganancias, Reino Unido. Mientras que las películas producidas en EEUU son cerca de 10 veces más que las del segundo país.

Esto también se refleja en el lenguaje de las películas, ya que de los 8 idiomas que más películas tienen el 65 % corresponde a las habladas en inglés, seguido, muy por detrás, por el francés con el 9 %



### Sobre la API:

La API fue realizada mediante programación en lenguaje Python. Este lenguaje cuenta con una serie de librerías y funciones que permiten generar las interacciones necesarias mediante una codificación sencilla. Las librerías utilizadas son las siguientes:

**Pandas:** Es una biblioteca de código abierto muy utilizada en la ciencia de datos para el análisis, manipulación y limpieza de datos. Utiliza los DataFrame como estructura de datos.

**NumPy:** Es una biblioteca fundamental para el cálculo científico en Python. Ofrece soporte para operaciones matemáticas y manipulación de matrices multidimensionales, siendo la base para muchas otras bibliotecas de ciencia de datos y aprendizaje automático.

**FastAPI:** Es un marco de desarrollo de aplicaciones web en Python que facilita la creación de APIs. Se utiliza junto con Uvicorn para ejecutar aplicaciones de manera local.

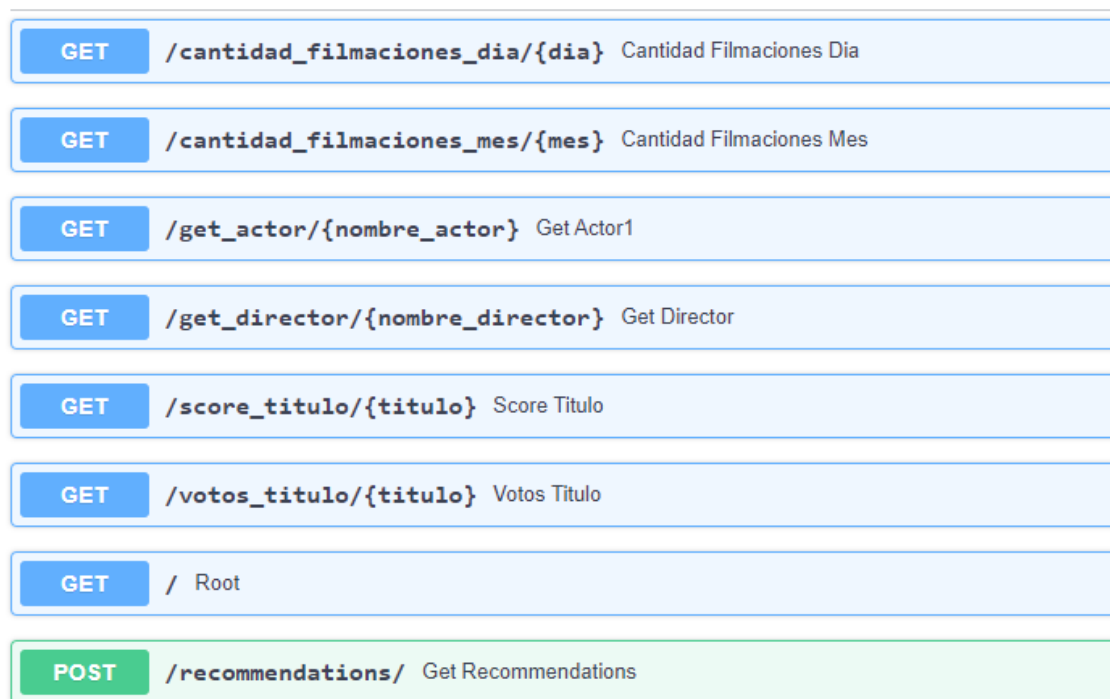
**Uvicorn:** Es un servidor ASGI (Asynchronous Server Gateway Interface) para Python que se utiliza para ejecutar aplicaciones.



Scikit-learn: Es una biblioteca de aprendizaje automático en Python que proporciona herramientas eficientes para tareas de análisis predictivo de datos. Está basada en NumPy, SciPy y Matplotlib.

SciPy: Es una biblioteca de Python para cálculos científicos y técnicos. Está construida sobre NumPy y ofrece módulos de optimización.

Luego de preparar los dataset, por ejemplo, tomando las fechas en formato YYYY-MM-DD y genere dos nuevas columnas una con el mes en español y otra con el día de la semana, también en español. Se generaron seis aplicaciones de consulta y una aplicación de post (ver ilustración 6) ya que en esta ultima los datos ingresados por el usuario forman parte del procesamiento para determinar la similitud con otras películas.



GET	/cantidad_filmaciones_dia/{dia}	Cantidad Filmaciones Dia
GET	/cantidad_filmaciones_mes/{mes}	Cantidad Filmaciones Mes
GET	/get_actor/{nombre_actor}	Get Actor1
GET	/get_director/{nombre_director}	Get Director
GET	/score_titulo/{titulo}	Score Titulo
GET	/votos_titulo/{titulo}	Votos Titulo
GET	/	Root
POST	/recommendations/	Get Recommendations

Ilustración 6: En la imagen, se muestra las aplicaciones generadas en Render

Veamos cómo funciona cada una:

@app.get("/cantidad\_filmaciones\_dia/{dia}") : Se le brinda un día de semana en español y devuelve la cantidad de películas estrenadas ese día. Por ejemplo: Lunes

@app.get("/cantidad\_filmaciones\_mes/{mes}") : En este caso la entrada es un el nombre de un mes es español y también devuelve la cantidad de películas que fueron estrenadas en ese mes. Por ejemplo: Agosto

@app.get("/get\_actor/{nombre\_actor}") : Se brinda el nombre de un actor y nos da la cantidad de películas en donde ese actor participó, la ganancia total y el promedio.

@app.get("/get\_director/{nombre\_director}") : Brindando el nombre de un director nos devuelve la cantidad de films dirigidos y un detalle de cada una de las películas. A continuación un ejemplo del detalle brindado de una película:

```
"titulo": "Alice in Wonderland",
"fecha_lanzamiento": "2010-03-03T00:00:00",
"retorno": 5.12745555,
```

```
"costo": 200000000,  
"ganancia": 1025491110
```

@app.get("/score\_titulo/{titulo}"): Ingresando el título de una película me devuelve en que año se estrenó y la ganancia que obtuvo.

@app.get("/score\_titulo/{titulo}"): También se ingresa el nombre de una película y nos devuelve el año de estreno, la cantidad de votos que obtuvo esta película y la valoración promedio. Si la película tiene una valoración total menor a los 2000, la API no realiza la consulta y nos avisa que es porque no cumple la condición.

@app.post("/recommendations/"): En este caso se nos pide el Título de una película y la cantidad de recomendaciones (n) que deseamos obtener. Y la aplicación realiza una consulta al dataset y nos devuelve n películas similares a la ingresada. Para determinar la similitud entre las películas se utiliza un modelo de Nearest Neighbors (NN) donde se tiene en cuenta el título de la película ingresada, por si forma parte de una saga, el resumen de la misma, la popularidad y la evaluación promedio. Se eligió NN para la aplicación final porque se quería trabajar con todo el dataset disponible. Ya que de haber elegido el método por matriz de similitud, por ejemplo, si bien los resultados mejorarían habría que hacer un sub sample del dataset para reducir los datos y no tener problemas de memoria.

Si se desea utilizar este método para la consulta, puedes clonar este repositorio de GitHub

[https://github.com/LauchaLogico/recomendaciones\\_peliculas.git](https://github.com/LauchaLogico/recomendaciones_peliculas.git)