Práctica 1: Tipología y Ciclo de vida de los datos

Películas ganadoras de premios Oscar

Clara Matilde Roca de la Concha

Victor H. Ayala Sánchez

Índice

Autores 2

1. Contexto 2

2. Título 2

3. Descripción del conjunto de datos 2

4. Representación gráfica 3

5. Contenido 4

6. Propietario 8

7. Inspiración 8

8. Licencia 9

9. Código 9

10. Dataset 11

11. Video 11

Contribuciones 11

Autores

- Clara Matilde Roca de la Concha
- Victor H. Ayala Sánchez

1. Contexto

La base de partida para nuestra araña es una tabla en Wikipedia que agrupa información sobre las diferentes películas que han ganado algún premio Oscar (en todas las categorías). Como veremos en el apartado 'Inspiración', esta tabla proporciona el punto de partida inicial para nuestra recopilación de datos.

Como sabemos, Wikipedia es la mayor enciclopedia virtual y proporciona un contexto ideal para poder trabajar en un marco colaborativo y para poder utilizar nuestra araña.

Una vez obtenida la información de esta tabla, hemos pasado a entrar en cada uno de los links de las películas para recolectar información como la dirección, reparto, distribución, presupuesto de la película, etc.

Hecho esto, hemos entrado a cada uno de los links de actores y actrices para obtener información bibliográfica: fecha y lugar de nacimiento, género, etc.

2. Título

El título del conjunto de datos es: **Películas ganadoras de premios Oscar**.

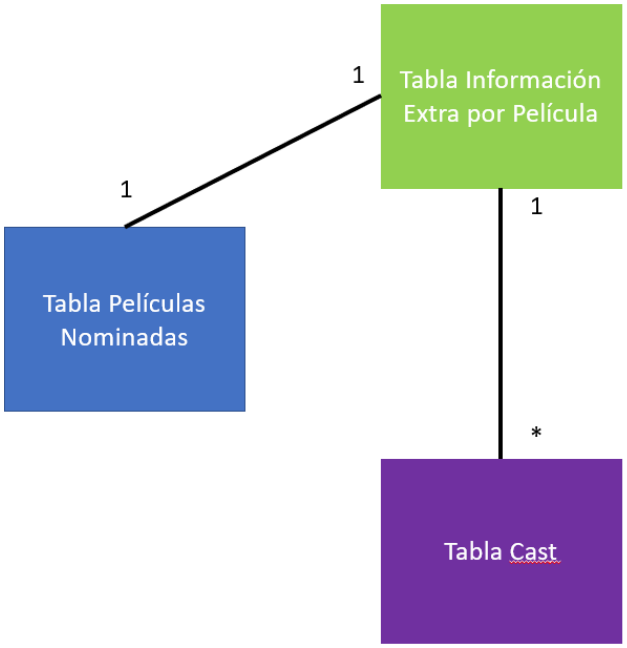
3. Descripción del conjunto de datos

El conjunto de datos se encuentra conformado por tres tablas: una tabla de películas nominadas (`df_movies.csv`), una segunda tabla con información extra de cada película (`df_metamovies.csv`) y la tercera que corresponde a la información de los actores y actrices que participan en cada película (`df_castData.csv`).

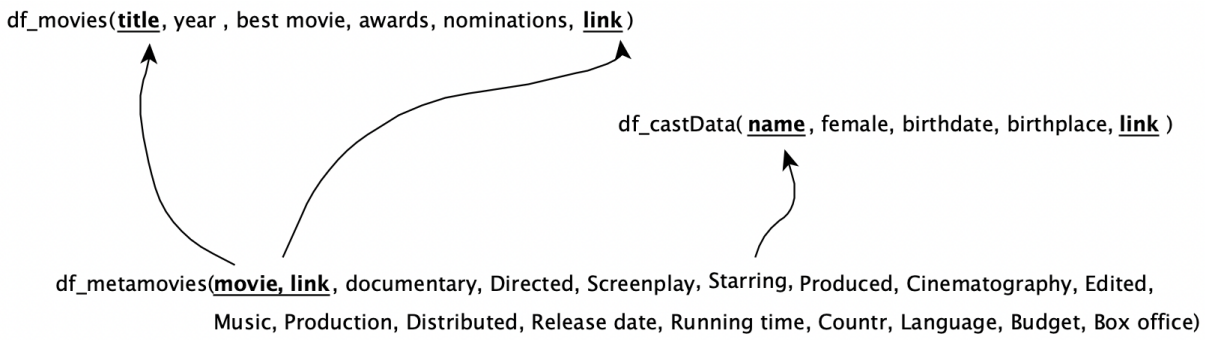
Entraremos en detalle sobre el contenido de cada uno de los dataset en el apartado 'Contenido'.

4. Representación gráfica

A continuación se presenta un diagrama de entidad relación con las tres tablas que componen el conjunto de datos:



La tabla de películas nominadas se relaciona 1 a 1 con la tabla que contiene la información extra y esta a su vez se relaciona 1 a muchos con la tabla de casting.



Como podemos observar, la forma de relacionarse de la tabla *df_movies* a la tabla *df_metamovies* es a través de *title-movie* y el link, mientras que la tabla *df_metamovies* y *df_castData* se relacionan vía *Starring-name*.

5. Contenido

Según lo ya comentado el conjunto de datos final se compone de tres tablas separadas:

df_movies: Información recolectada de la primera tabla.

Observaciones generales:

- Tamaño: 1348 filas × 6 columnas
- Eliminar la primera fila, que es nula.
- El campo linkMovie permite reconstruir una key que relaciona con la tabla de información extra.

Nombre de campo	Descripción	Tipo	Ejemplo	Observaciones
title	Título de la película	Texto	Dune	
year	Año de estreno	Numérico	2021	1 valor nulo.
best movie	Determina si la película ganó o no el Oscar a Mejor Película	Booleano	False	True = Mejor Película False = No ganó Mejor Película
awards	Número de premios Oscar	Numérico	5	
nominations	Número de nominaciones	Texto	3\n	Será necesaria limpieza: eliminar \n y convertir en numérico.
linkMovie	Parte del link de cada una de las películas	Texto	/wiki/CODA_(2021_film)	Para acceder, hay que añadir el enlace de la página principal: https://en.wikipedia.org/

df_metamovies: Información recolectada a partir de cada uno de los links de la columna linkMovie de la tabla df_movies.

Observaciones generales:

- Tamaño: 1346 filas × 18 columnas
- Homogeneizar nombres de columnas (inicio mayúsculas/minúsculas,...).
- Reordenar las columnas, para tener la información más relevante primero (i.e., poner los links al final).
- Hemos incluído categoría *documental*, ya que consideramos importante separar este tipo de películas, ya que suelen tener un formato distinto (no tiene guión ni reparto al uso).

- Las categorías de la ficha técnica pueden contener más de un elemento en cada celda.
- Algunas categorías como *Country* fueron cortadas al final para que la búsqueda encontrara también su versión en plural.

Nombre de campo	Descripción	Tipo	Ejemplo	Observaciones
movie	Título de la película	Texto	Dune	
link	Link de cada una de las películas	Texto	https://en.wikipedia.org/wiki/CODA_(2021_film)	
documentary	Determina si la película es documental o no	Booleano	True	True = documental False = ficción
Directed	Director/a	Texto (lista)	[Sian Heder]	Puede contener más de un nombre. Celdas vacías = 21
Screenplay	Guionista	Texto	[Sian Heder]	Puede contener más de un nombre. Celdas vacías (documental = False) = 552
Starring	Reparto	Texto	[Emilia Jones, Eugenio Derbez, Troy Kotsur]	Puede contener más de un nombre. Celdas vacías (documental = False) = 55
Produced	Productor/a	Texto	[Geoff McLean]	Puede contener más de un nombre. Celdas vacías = 43
Cinematography	Director/a de fotografía	Texto	[Paula Huidobro]	Puede contener más de un nombre. Celdas vacías = 186
Edited	Editor/a	Texto	[Geraud Brisson]	Puede contener más de un nombre.

Nombre de campo	Descripción	Tipo	Ejemplo	Observaciones
				Celdas vacías = 158
Music	Compositor/a	Texto	[Marius de Vries]	Puede contener más de un nombre. Celdas vacías = 214
Production	Compañía de producción	Texto	['Fox Searchlight Pictures', 'TSG Entertainment', 'Double Dare You Productions']	Cambiar nombre de columna a: “Production company”. Puede contener más de un nombre. Celdas vacías = 348
Distributed	Distribuidora	Texto	[Apple Studios]	Puede contener más de un nombre. Celdas vacías = 77
Release date	Fechas de estreno	Texto	[January 28, 2021, August 13, 2021]	Es necesario limpiar este campo y convertirlo en tipo Date. Celdas vacías = 13
Running time	Duración	Texto	[111 minutes]	Interesante para filtrar cortometrajes de largometrajes. Limpiar y cambiar el formato a ‘time’ o ‘numeric’. Celdas vacías = 21
Countr	País de origen	Texto	[United States]	Cambiar nombre de la columna a: “Country”. Puede contener más de un nombre. Celdas vacías = 31

Nombre de campo	Descripción	Tipo	Ejemplo	Observaciones
Language	Idioma original	Texto	[American Sign Language, English]	Celdas vacías = 46
Budget	Presupuesto	Texto	[$10 million]	Limpiar y pasar a numérico. Celdas vacías = 516
Box Office	Taquilla	Texto	[$1.6 million]	Limpiar y pasar a numérico. Celdas vacías = 431

df_castData: Información recolectada a partir de cada uno de los links del reparto de cada película.

Observaciones generales:

- 3255 filas × 5 columnas
- Analizar y eliminar 17 duplicados
- Se relaciona con la tabla de películas

Nombre de campo	Descripción	Tipo	Ejemplo	Observaciones
name	Nombre de actor/actriz	Texto	Emilia Jones	
female	Determina si es hombre o mujer	Texto	True	True = mujer False = hombre Valores nulos ('NA') = 5
birthdate	Fecha de nacimiento	Fecha	2002-02-23	Valores nulos = 193
birthplace	Lugar de nacimiento	Texto	[Mesa, , , Arizona, , U.S.]	Limpiar. Valores nulos = 215
link	Enlace de la página del	Texto	https://en.wikipedia.or	

Nombre de campo	Descripción	Tipo	Ejemplo	Observaciones
	actor/actriz		g/wiki/Emilia_Jones	

6. Propietario

Los datos obtenidos en el proyecto se encuentran publicados en Wikipedia, que al tratarse de un proyecto de enciclopedia libre, es información aportada por la comunidad; por lo tanto no hay un propietario definido.

Se pueden encontrar conjuntos de datos similares en Kaggle, aportados por diferentes colaboradores de la comunidad, como por ejemplo:

Fontes, R. The Oscar Award, 1927 - 2020. Kaggle: https://www.kaggle.com/datasets/unanimad/the-oscar-award

En el caso comentado, el autor del conjunto de datos hizo scraping de la información directamente desde la página oficial de los premios de la Academia. Sin embargo, en este caso se recopilan nominados y ganadores de todas las categorías, pero no se profundiza en mayores detalles como es el caso de nuestras tablas, que incluye fechas de nacimiento, género, presupuesto y otros múltiples campos.

Tampoco se comentan aspectos éticos y legales dentro del proyecto mencionado. En el caso del presente proyecto, nos aseguramos de analizar la página web antes de realizar pruebas de scraping, comprobando que el uso de arañas estaba permitido (analizando el robots.txt) y que la información a obtener es de dominio público.

7. Inspiración

A raíz de la polémica con la edad de las novias de Leonardo DiCaprio, surgió la pregunta si esta costumbre del actor por elegir a chicas menores de 25 estaba vinculada a una percepción de que la juventud en las mujeres es mucho más importante que en los hombres. En este contexto, queríamos averiguar si los roles principales de las películas de Hollywood también se veían afectados por este sesgo.

¿Son las actrices escogidas en esta películas más jóvenes que sus contrapartes masculinas? Para ello elegimos una muestra de cerca de 1350 películas: las películas premiadas en los Óscar.

Nos parece una muestra interesante, ya que estos premios son los más importantes en la cinematografía estadounidense e implican, necesariamente, que estas películas tengan un reconocimiento y una visibilidad notable en el mundo del cine.

En un primer momento íbamos a partir de la lista de las 250 películas mejor valoradas de IMDb, sin embargo, nos pareció más interesante el desafío de bucear por la información de 1346 películas, la extensión de la tabla de películas de Wikipedia.

Encontramos varios datasets de películas y ganadores o nominados a los premios Óscar, si bien no encontramos una tabla que reuniera tanto películas ganadoras como detalles del elenco que las protagoniza. Por ello, creemos que nuestro conjunto de datos tiene valor y sirve un propósito concreto.

8. Licencia

Hemos escogido este tipo de licencia, para que la información contenida en el proyecto pueda ser compartida bajo cualquier formato dentro de la comunidad de desarrolladores, científicos y analistas de datos, sin problemas o limitaciones. Al tratarse de información de dominio público, a la que sólo hemos accedido y organizado en ficheros, creemos que debe ser utilizada con fines académicos y por lo tanto no comerciales.

9. Código

El código fuente en Python puede ser consultado en el repositorio de github.

El proyecto se encuentra conformado por un directorio raíz y dos subdirectorios. En el subdirectorio `source` se tiene un fichero notebook con la spyder y su correspondiente log de ejecución.

El resultado de ejecución de la araña se guarda en el fichero `data_wiki.json`. Además de los mencionados ficheros, en el sub directorio source también se encuentra un fichero .py con el mismo código del notebook, para ejecutar la araña por consola de ser necesario.

En el directorio raíz se tiene otro fichero notebook llamado `Dataclean`. El objetivo de este es tomar como entrada el fichero `data_wiki.json`, separar la información en las tres tablas que conforman el conjunto de datos y guardarla en tres ficheros csv. El resultado de ejecutar el notebook de limpieza se puede encontrar en el subdirectorio `dataset`, con los ficheros correspondientes a las películas, su información extra y la tabla de actores y actrices.

Además de lo comentado anteriormente, en el directorio raíz, también se tiene un fichero .txt con las librerías necesarias para la ejecución del código y un fichero `readme` con la misma descripción comentada.

A la hora de escribir el código, los desafíos más relevantes a los que nos hemos enfrentado han sido, en primer lugar, el de promover un uso responsable de la spider para no saturar el servidor, tal como se requiere en el archivo robots.txt de Wikipedia.

Sabíamos que íbamos a bucear a lo largo de casi 5.000 páginas, para extraer una serie de datos, por lo que era importante establecer ciertas limitaciones. Para ello, se implementaron tiempos de espera aleatorios para evitar realizar un número elevado de peticiones en un corto período de tiempo.

Otra de las medidas para evitar bloqueos dentro de la spyder ha consistido en definir un *user agent* dentro de los parámetros de ejecución (`'USER_AGENT': 'Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)`) con el objetivo de simular el comportamiento de una persona en la navegación de los diferentes links.

Otro de los desafíos a los que nos enfrentamos fue el de designar el género de los actores y actrices, ya que esta información no se encontraba en forma de tabla por la que pudiéramos programar una extracción simple. Para ello, en primer lugar le pedimos a la spider que analizara las categorías de cada actor y actriz. Si había palabras como 'male' o 'actor', categorizamos a la persona como hombre (`female: False`). En caso de encontrar palabras como 'female' o 'actress', la categorizamos como mujer (`female: True`).

Ejemplo de categorías que aparece en la página del actor Brad Pit en wikipedia.org

En caso de que estas categorías no existan en ese determinado perfil, o que no den un resultado fructífero, la función pasa a analizar el texto de cada enlace y cuenta las palabras masculinas ('he', his', 'man', …) y las femeninas ('she', 'her', 'woman',...) para luego compararlas. Si el resultado son más palabras masculinas: la persona se cataloga como hombre; en caso contrario, se ha catalogado como mujer.

Por último, queremos aclarar que somos conscientes de que la información de la primera tabla podía obtenerse con una simple línea en `pandas`, de la forma `pd.read_html('url')`, pero hemos decidido iniciar la recolección con la spider, para usar al máximo este recurso.

10. Dataset

El conjunto de datos se encuentra tanto en repositorio de github como en el siguiente enlace de zenodo. El enlace DOI de la publicación en Zenodo es:
https://doi.org/10.5281/zenodo.7328744

11. Video

En este enlace se puede encontrar el vídeo de la práctica.

https://drive.google.com/file/d/1Ik0g9dRlF97nCpDXFV7j4Q_0oKdwKSxF/view?usp=share_link

Contribuciones

Contribuciones	Firma
Investigación previa	V. H. A., C.R.C.
Redacción de las respuestas	V. H. A., C.R.C.
Desarrollo del código	V. H. A., C.R.C.
Participación en el vídeo	V. H. A., C.R.C.