Informe Movie Data
Proyecto Analítico de Datos

Profesor
Mario Ortiz

Alumno
Johan Órdenes
Leonel Villagra

Fecha
27/10/2020

Índice

Análisis de los datos **3**

Limpieza de los datos **4**

Comparación de datos **6**

Tiempo de compilación **6**

Exportación de archivos **7**

Carga a la bases de datos **8**

En esta investigación se nos pide hacer el proceso de ETL, es decir exportar, transformar y cargar los datos del archivo moviedata.xlsx. Esto lo hicimos realizando la lectura de los datos entregados, limpieza y eliminación de los datos inservibles, y por último el almacenamiento de los datos en una base de datos en Postgres. Para esto separamos nuestro código en dos fases. La primera para la lectura y guardado de archivos que veremos más adelante, y la segunda para subir netamente dichos archivos a la base de datos.

1. Análisis de los datos

color	director_name	num_critic_for_reviews	duratio	director_facebook_likes	actor_3_facebook_likes	actor_2_name	actor_1_facebook_likes	gross
Color	Mikael HÃ¥fstrÃ¶m	286	115	101	585	50 Cent	13000	25121291
Color	Brian A Miller	46	93	32	1000	50 Cent	13000	
Color	Antoine Fuqua	305	124	845	424	50 Cent	15000	52418902
Color	Don Coscarelli	81	97	152	135	A. Michael Baldwin	674	7282851
Color	Randall Rubin	23	99	0	131	A.J. Buckley	363	
Color	Bobby Roth	1	120	40	249	Aaron Ashmore	912	
Color	Alejandro AmenÃ¡bar	140	106	448	648	Aaron Ashmore	9000	54734
Color	John Dahl	121	93	131	90	Aaron Hughes	497	2426851
Color	Pou-Soi Cheang	14	119	3	22	Aaron Kwok	879	
Color	Gary Winick	91	78	56	184	Aaron Stanford	376	2882062
Color	Kirsten Sheridan	148	114	59	403	Aaron Staton	49000	31655091
Color	D.J. Caruso	253	105	154	501	Aaron Yoo	884	80050171
Color	D.J. Caruso	253	105	154	502	Aaron Yoo	884	80050171
Color	Jonathan Levine	147	99	129	362	Aaron Yoo	976	2077046
Color	Larry Charles	313	99	119	174	Aasif Mandvi	600	59617068
Color	Shekhar Kapur	202	114	159	877	Abbie Cornish	13000	16264475
Color	Zack Snyder	435	128	0	826	Abbie Cornish	4000	36381716
Color	JosÃ© Padilha	492	117	294	1000	Abbie Cornish	10000	58607007
Color	JosÃ© Padilha	492	117	294	1000	Abbie Cornish	10000	58607007
Color	R. Balki	12	133	12	106	Abhishek Bachchan	464	199228
Color	Michael Apted	116	116	150	956	Abigail Spencer	18000	6002756
Color	McG	241	103	368	520	Abigail Spencer	27000	54758461
Color	Shawn Levy	156	103	189	949	Abigail Spencer	2000	34290142
Color	Sam Weisman	53	92	39	458	Abraham Benrubi	3000	105263257
Color	Craig Gillespie	178	117	44	531	Abraham Benrubi	788	27550735
Color	Eric England	15	81	15	93	Ace Marrero	847	
Color	Frank Oz	140	85	0	490	Adam Alexi-Malle	21000	66365290
Color	Steve Rash	13	98	15	281	Adam Arkin	691	8500000
Color	Ben Lewin	337	95	20	365	Adam Arkin	422	5997134
Color	Roland Emmerich	185	154	776	933	Adam Baldwin	10000	306124059
Color	Roland Emmerich	192	142	776	1000	Adam Baldwin	13000	113330342
Color	Chris Robinson	54	105	49	104	Adam Boyer	680	21160089
Color	Peter Jackson	645	182	0	773	Adam Brown	5000	303001229
Color	Peter Jackson	509	186	0	773	Adam Brown	5000	258355354
Color	Peter Jackson	422	164	0	773	Adam Brown	5000	255108370
Color	Michael McGowan	47	98	12	149	Adam Butcher	393	795126

Tabla de datos “moviedata.xlsx”
Fuente: “Escuela de Ingeniería UCN”

Las celdas vacías y erróneas contenían los siguientes N/A, #ERROR, #N/I, ?, #VALOR!, #NODATA. Lo que decidimos hacer en este caso fue:

- Pasar datos tipo texto no conocidos como “UNKNOWN”.
- Pasar datos tipo número no conocidos como “0”.
- “?,-1,0” no se eliminaron, dado que no afectaba a los datos el cambiarlos.

2. Limpieza de los datos

También añadimos funciones que nos sirven para la limpieza luego de leer los datos de cada columna, definiendo los valores NULL como ‘nan’ que son las siguientes:

- cleanColor(): Limpia los datos de la columna “color”
  - Reemplaza el string ‘ERROR’ por UNKNOWN
  - Reemplaza el cast ‘nan’ por UNKNOWN
  - Corta los espacios antes de que empiece el string.

- cleanName(): Limpia los datos de las columnas tipo texto.
  - Reemplaza el string ‘#NODATA’ por NULL
  - Reemplaza el cast ‘nan’ por UNKNOWN
  - Reemplaza el string ‘?’ por UNKNOWN
  - Corta los espacios antes de que empiece el string y transforma los errores traduciendolos a carácteres latinos en UTF-8.

- cleanMovie(): Limpia los datos de la columna “title_movie”.
  - Elimina el último caracter ‘Â’ del string

- cleanDigit(): Limpia los datos de las columnas que contienen dígitos enteros.
  - Reemplaza el string ‘#NODATA’ por UNKNOWN
  - Reemplaza el string ‘nan’ por UNKNOWN
  - Reemplaza el string ‘N/I’ por UNKNOWN

- cleanDecimal(): Limpia los datos de las columnas que contienen dígitos flotantes.
  - Reemplaza el string ‘nan’ por UNKNOWN

También hubo columnas que no se limpiaron, esto fué porque los datos están 100% funcionales y sin problemas para un posterior análisis y agregación a la base de datos, las columnas son las siguientes:

- genres
- num_voted_users
- cast_total_facebook_likes
- movie_imdb_link
- language

- country
- content_rating
- budget
- title_year
- imdb_score
- movie_facebook_likes

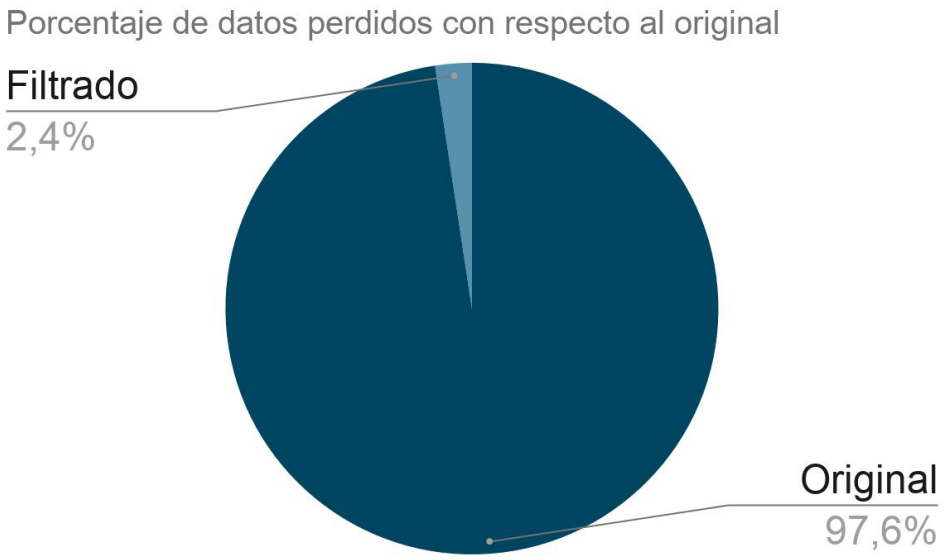
Además hubo columnas que fusionamos, en este caso “num_voted_users” y “cast_total_facebook_likes” para una película en particular tenía una variación muy pequeña de votos en aproximadamente 2 o 3 votos por película, por lo que deducimos que no eran datos altamente representativos y no generaban un cambio al momento de agregarla a la base de datos, esto nos ayuda a que haya un ahorro de cantidad de datos ingresados.

movie_title	num_voted_users	cast_total_facebook_likes
DisturbiaÂ	186984	2287
DisturbiaÂ	186982	2288
The WacknessÂ	27266	2748
The DictatorÂ	213863	1375
Elizabeth: The Golden AgeÂ	54787	16899
Sucker PunchÂ	197584	7067
RoboCopÂ	182899	14161
RoboCopÂ	182910	14160

Lo que decidimos en este caso fue obtener un promedio entre las filas con películas del mismo nombre y reducirlas a una sola fila.

3. Comparación de datos

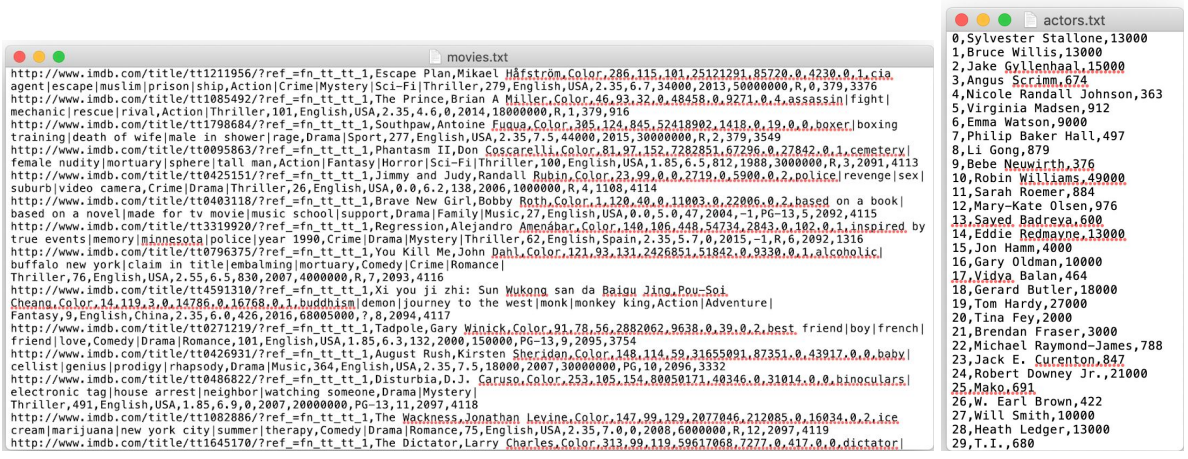
Luego de hacer toda la limpieza de datos anterior, decidimos comparar los datos que teníamos al principio, con los datos nuevos dentro del mismo dataframe modificado y llegamos a la siguiente conclusión:



Cantidad de datos Originales vs Cantidad de datos filtrados
Fuente: Elaboración de Equipo

La cantidad de datos originales era de 5.043 columnas, mientras que luego de haber limpiado y filtrado todos los datos quedaron, disminuyó la cantidad de filas que teníamos al principio a 4.919. Esto significa que hubo un 2,4% de pérdida de información.

4. Exportación de archivos



Archivos finales del proyecto
Fuente: Elaboración propia

Aquí se tienen todos los archivos que se generan luego de ejecutar el código principal. Se generan los archivos “movies.txt” y “actors.txt” para su posterior carga a la base de datos. Lo hicimos de esta manera (Importación-Exportación) para evitar la espera que se procesen los datos mientras hacemos pruebas al programar. Esto puede mejorar haciendo uso de hilos.

5. Carga a la bases de datos

Finalmente para la elección de tablas decidimos crear 2 tablas mostradas a continuación:

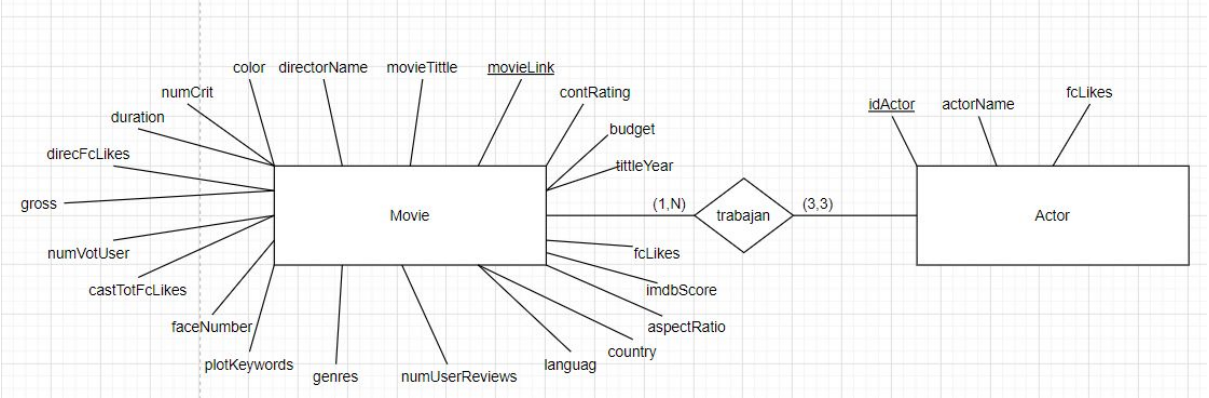


Diagrama Entidad-Relacionamiento
Fuente: Elaboración de Equipo