Universidad de Sonora

Introduction to Data Science and its Methodologies

Final Project:

Creation and Management of Databases

Ana Daniela Pérez Romero / A222230117

Professor: Juan Pablo Soto Barrera

Hermosillo, Sonora 03/12/2022

Introduction

Introduction to Data Science and its Methodologies is a course focused on teaching the most common methodologies used in any data science scenario. Throughout the course, it is taught how to ensure that the data used in problem solving is relevant and properly manipulated. Accordingly, we learned how to form a business/research problem, collecting, preparing, and analyzing data, applying the 6 stages of the CRISP-DM methodology, working, and creating databases with SQL, but most importantly, how to make use of these tools and knowledge altogether to develop hands-on experience.

The final project for the second phase of the course consists of developing an application that allows analyzing information contained in a SQL database. This database will be later consulted though a Jupyter notebook and should be analyzed through graphs and tables to solve a problem. Within the requirements of the database created through SQL, is that it contains at least one view, one stored procedure and one function.

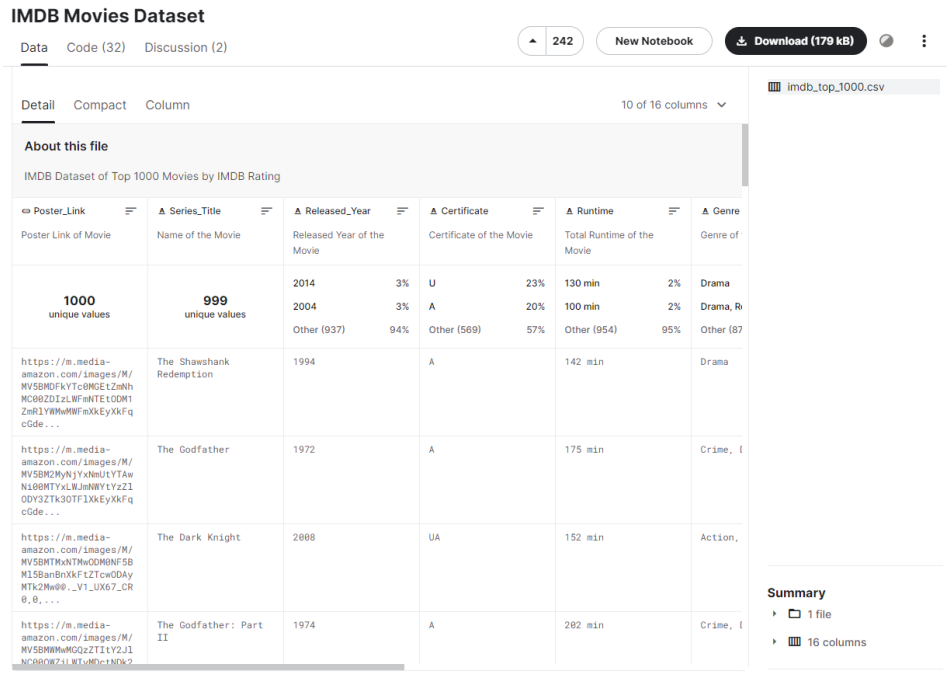
Project Documentation

I. Database Creation

For the development of this project, I decided to work on one of my greatest passions: watching movies. Perhaps, you are wondering why this topic is so important for me, and the truth is that as a full-time employee and student, I hardly find free time to do those activities that I enjoy the most. I usually watch movies Sunday night, and every time I spend at least 20 minutes just deciding what movie should I watch; there are so many options that I just get overwhelmed. On the other hand, there have been times where friends and family recommend me movies and turns out they do not coincide with my cinematographic taste, meaning I just lost two hours of my short weekend. As a Data Science student, I feel the moral responsibility to apply what I have learned in class to simplify my life; that is one of Data Analytics goals right?

The first stage of any Data Science project is obtaining data, and based on the subject of my project I decided to use a dataset from Internet Movie Database (IMDb), where they display the top 1000 best ranked movies:

https://www.kaggle.com/datasets/harshitshankhdhar/imdb-dataset-of-top-1000-movies-and-tv-shows



Through MySQL a new schema was created and called “movies”; firstly, it was supposed that the table was going to be created from Table Data Import Wizard option, however it resulted in missing records. For this reason, the table was created through a Jupyter Notebook, creating a data frame with pandas, and using sqlalchemy library to upload the data frame to MySQL.

```
import pandas as pd
from sqlalchemy import create_engine
import pymysql
```

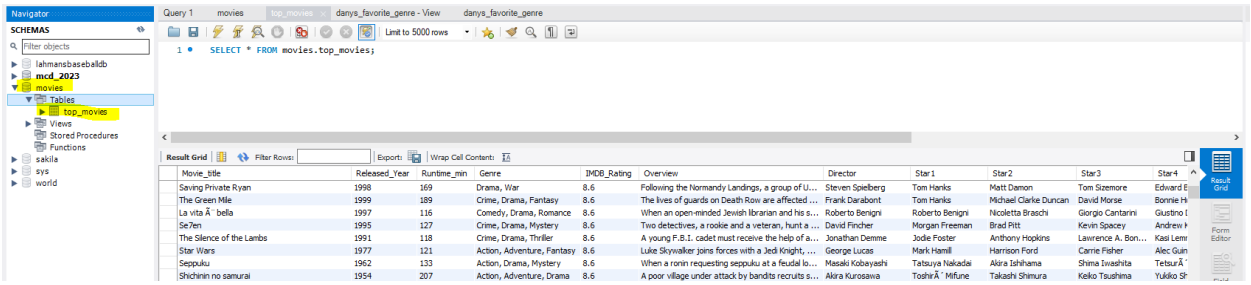
```
# Creation of dataframe from a .csv file
movies = pd.read_csv("top_imdb.csv")
```

```
# We store the necessary information to access MySQL
myUser = 'root'
myPass = '[redacted]'
myEndpoint = 'localhost'
myPort = 3306
myPort2 = ':' + str(myPort) + '/'
myDb = 'movies'

myDataConnect = [myUser, myPass, myEndpoint, myPort, myDb]
```

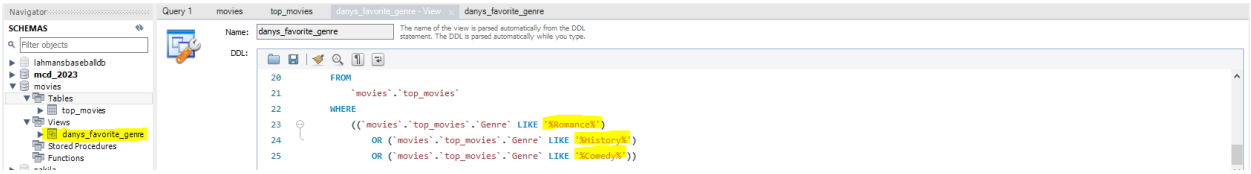
```
db_data = 'mysql+mysqldb://' + myUser + ':' + myPass + '@' + myEndpoint + myPort2 \
       + myDb
engine = create_engine(db_data)
```

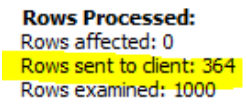
```
# Execute the to_sql for writting DF into MySQL
movies.to_sql('top_movies', engine, if_exists='replace', index=False)
```

Once the program is executed, the table “top_movies” is uploaded to MySQL’s schema “movies”. With *SELECT * FROM movies.top_movies* query, 1000 records should be displayed.

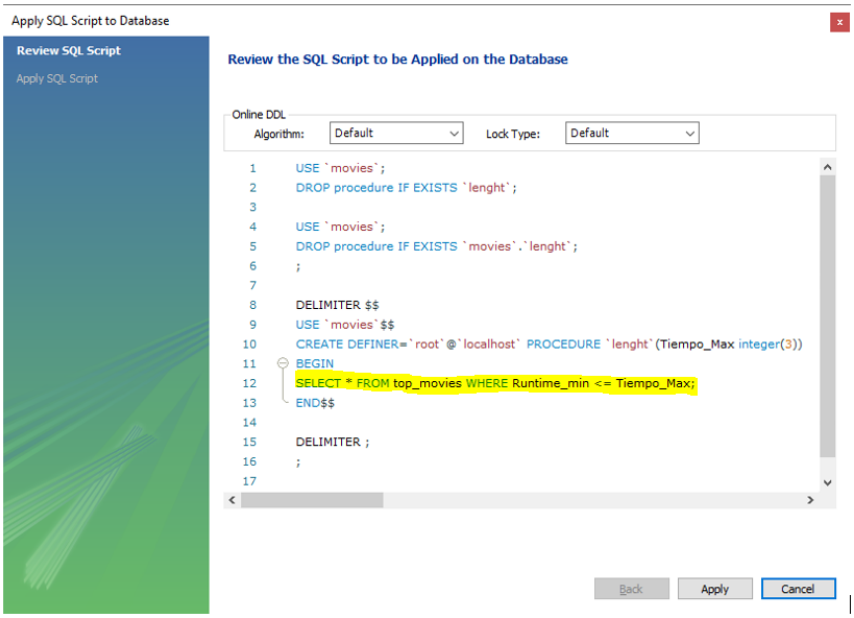


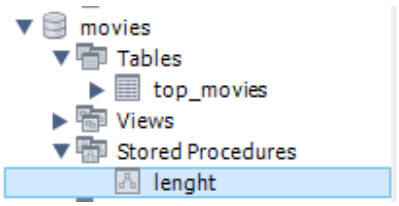
Data is uploaded, however, there are still a thousand movies that need to be filtered based on my cinematographic taste. A new view called “danys_favorite_genre” is created selecting my favorite movie genres: romance, history, and comedy. With this view 636 records are left behind, leaving only 364 movies to be furthered analyzed.

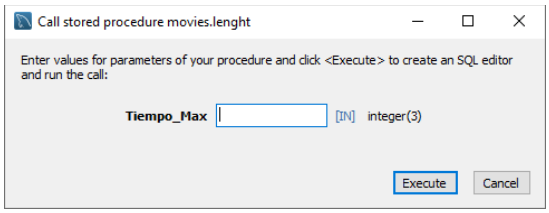




Another important factor when selecting a film is its length; there are days with less time availability, where shorter movies must be selected. A stored procedure “length” was created and allows to filter movies based on the maximum time in minutes that can be consigned to watch them. This procedure allows the user to type a maximum number of minutes long for a movie and return movies equal or less than the number typed.








```
call movies.lenght(95);
```

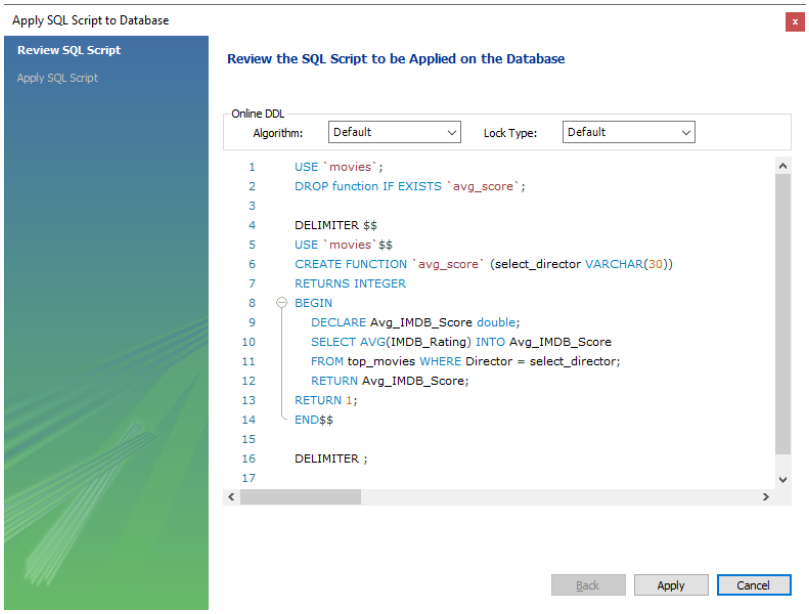
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

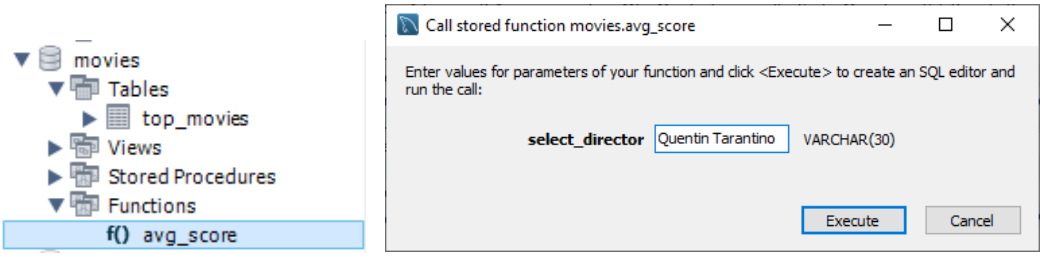
Movie_title	Released_Year	Runtime_min	Genre	IMDB_Rating	Overview	Director	Star1	Star2	Star3
The Lion King	1994	88	Animation, Adventure, Drama	8.5	Lion prince Simba and his father are targeted b...	Roger Allers	Rob Minkoff	Matthew Broderick	Jeremy Irons
Hotaru no haka	1988	89	Animation, Drama, War	8.5	A young boy and his little sister struggle to surv...	Isao Takahata	Tsutomu Tatsumi	Ayano Shiraishi	Akemi Yamaguchi
Modern Times	1936	87	Comedy, Drama, Family	8.5	The Tramp struggles to live in modern industrial ...	Charles Chaplin	Charles Chaplin	Paulette Goddard	Henry Bergman
City Lights	1931	87	Comedy, Drama, Romance	8.5	With the aid of a wealthy erratic tippler, a dewy...	Charles Chaplin	Charles Chaplin	Virginia Cherrill	Florence Lee
Dr. Strangelove or: How I Learned to Stop Wor...	1964	95	Comedy	8.4	An insane general triggers a path to nuclear hol...	Stanley Kubrick	Peter Sellers	George C. Scott	Sterling Hayden
Paths of Glory	1957	88	Drama, War	8.4	After refusing to attack an enemy position, a g...	Stanley Kubrick	Kirk Douglas	Ralph Meeker	Adolphe Menjou
Bacheha-Ye aseman	1997	89	Drama, Family, Sport	8.3	After a boy loses his sister's pair of shoes, he g...	Majid Majidi	Mohammad Amir Naji	Amir Farrokh Hashemian	Bahare Seddiqi
Toy Story	1995	81	Animation, Adventure, Comedy	8.3	A cowboy doll is profoundly threatened and jeal...	John Lasseter	Tom Hanks	Tim Allen	Don Rickles
Ladri di biciclette	1948	89	Drama	8.3	In post-war Italy, a working-class man's bicycle ...	Vittorio De Sica	Lamberto Maggiorani	Enzo Staiola	Lianella Carell
The Kid	1921	68	Comedy, Drama, Family	8.3	The Tramp cares for an abandoned child, but e...	Charles Chaplin	Charles Chaplin	Edna Purviance	Jackie Coogan

For the creation of functions, a global variable must be defined: *log_bin_trust_function_creators = 1*; by default, it comes to 0. Once the global variable is defined, we can create functions

```
SET GLOBAL log_bin_trust_function_creators = 1;
```

An important aspect while selecting movies is its director; how are their films ranked? To answer this question, a function that returns the average rating per director was created.





"Quentin Tarantino" was selected as a director, once the function was runed, it returned an average score value of his movies of 8.175.

	movies.avg_score('Quentin Tarantino')
▶	8.175

II. Data Analysis through Python

Through a Jupyter Notebook, a connection is created to call the database "movies" from MySQL, which is set as a data frame.

```
conection=pymysql.connect(host= myEndpoint,
                          port= myPort,
                          user=myUser,
                          password= myPass,
                          db= myDb)
```

```
dataframe=pd.read_sql_query("SELECT * FROM movies.top_movies", conection)

dataframe.head(8)
```

	Movie_title	Released_Year	Runtime_min	Genre	IMDB_Rating	Overview	Director	Star1	Star2	Star3	Star4	No_of_Votes	Gross
0	The Shawshank Redemption	1994	142	Drama	9.3	Two imprisoned men bond over a number of years...	Frank Darabont	Tim Robbins	Morgan Freeman	Bob Gunton	William Sadler	2343110	28,341,469
1	The Godfather	1972	175	Crime, Drama	9.2	An organized crime dynasty's aging patriarch t...	Francis Ford Coppola	Marlon Brando	Al Pacino	James Caan	Diane Keaton	1620367	134,966,411
2	The Dark Knight	2008	152	Action, Crime, Drama	9.0	When the menace known as the Joker wreaks havo...	Christopher Nolan	Christian Bale	Heath Ledger	Aaron Eckhart	Michael Caine	2303232	534,858,444

Firstly, I would like to know if cinematography has been evolving positively or negatively over the years; for this matter, a table and a top 15 values bar graph showing the average IMDB Score Rate was created.

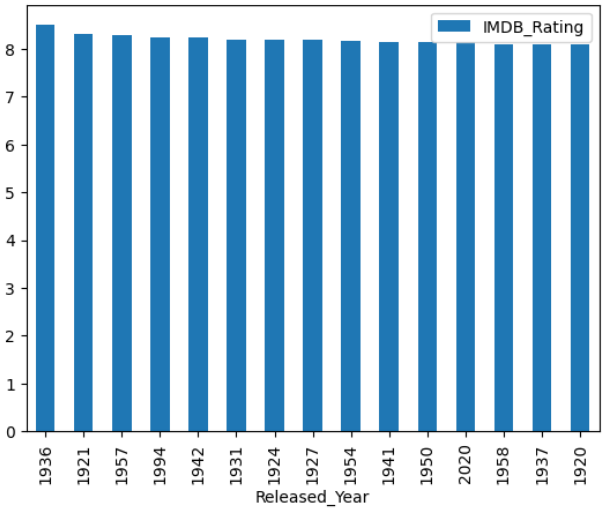
```
score_year= dataframe.groupby("Released_Year").mean()["IMDB_Rating"].reset_index()

score_year
```

```
scoreyear_pbar = score_year.sort_values('IMDB_Rating', ascending=False).head(15)
scoreyear_pbar
```

```
scoreyear_pbar.plot.bar(x="Released_Year",y="IMDB_Rating")
```

	Released_Year	IMDB_Rating
0	1920	8.100000
1	1921	8.300000
2	1922	7.900000
3	1924	8.200000
4	1925	8.100000
...	...	...
94	2016	7.935714
95	2017	7.890909
96	2018	7.994737
97	2019	7.995652
98	2020	8.133333

	Released_Year	IMDB_Rating
14	1936	8.500000
1	1921	8.300000
35	1957	8.277778
72	1994	8.238462
20	1942	8.233333
9	1931	8.200000
3	1924	8.200000
6	1927	8.200000
32	1954	8.166667
19	1941	8.150000
28	1950	8.140000
98	2020	8.133333
36	1958	8.100000
15	1937	8.100000
0	1920	8.100000



Apparently, it seems like the 21st century has not been the best cinematographic era; 14 out of the 15 best positions based on average score rate are taken by the 20th century. However, this information seems intriguing, so I decide to count how many films were evaluated per year. My suspicions where true: the number of films reviewed from the 20th century were smaller compared to those of the 21st century.

	Released_Year	IMDB_Rating
0	1920	1
1	1921	1
2	1922	1
3	1924	1
4	1925	2
...	...	...
94	2016	28
95	2017	22
96	2018	19
97	2019	23
98	2020	6

	Released_Year	IMDB_Rating
92	2014	32
82	2004	31
87	2009	29
94	2016	28
91	2013	28
79	2001	27
84	2006	26
85	2007	26
93	2015	25
90	2012	24
88	2010	23
71	1993	23
97	2019	23
81	2003	22
95	2017	22

Thanks to this analysis it is known that selecting a movie based on the average IMDB Score Rate from its released year is not the best method.

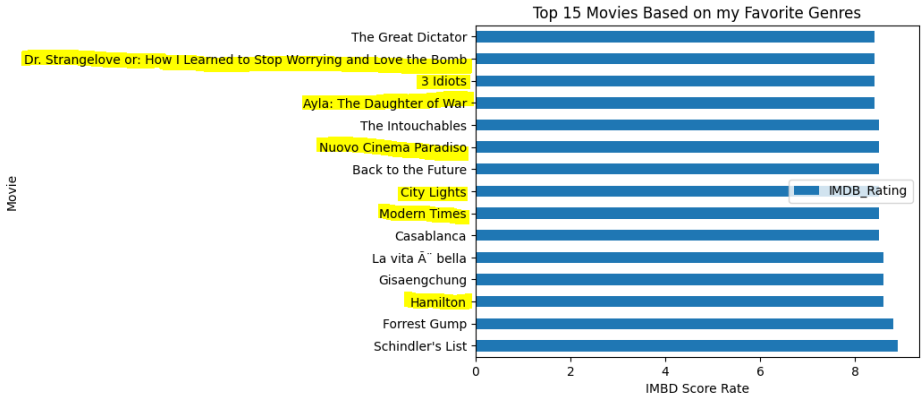
For a second analysis, I'd like to know which movies has the best IMDB score rate. A query was executed to filter the movies based on my favorite genres: Romance, History and Comedy. These values where saved in a data frame, that will later be used to create a plot bar that shows top 15 best ranked movies.

```
query='''SELECT * FROM movies.top_movies
        WHERE
        ((`movies`.`top_movies`.`Genre` LIKE '%Romance%')
            OR (`movies`.`top_movies`.`Genre` LIKE '%History%')
            OR (`movies`.`top_movies`.`Genre` LIKE '%Comedy%'))'''
```

```
dfFav=pd.read_sql_query(query, conection)
dfFav.head(8)
```

	Movie_title	Released_Year	Runtime_min	Genre	IMDB_Rating	Overview	Director	Star1	Star2	Star3	Star4	No_of_Votes	Gross
0	Schindler's List	1993	195	Biography, Drama, History	8.9	In German-occupied Poland during World War II,...	Steven Spielberg	Liam Neeson	Ralph Fiennes	Ben Kingsley	Caroline Goodall	1213505	96,898,818
1	Forrest Gump	1994	142	Drama, Romance	8.8	The presidencies of Kennedy and Johnson, the e...	Robert Zemeckis	Tom Hanks	Robin Wright	Gary Sinise	Sally Field	1809221	330,252,182
2	Hamilton	2020	160	Biography, Drama, History	8.6	The real life of one of America's foremost fou...	Thomas Kail	Lin-Manuel Miranda	Phillipa Soo	Leslie Odom Jr.	RenÃ©e Elise Goldsberry	55291	None

```
count_year_pbar.plot.barh(x="Movie_title",y="IMDB_Rating")
plt.title("Top 15 Movies Based on my Favortie Genres")
plt.xlabel("IMBD Score Rate")
plt.ylabel("Movie")
```



From the results above, and removing the movies that I have already seen in the past, I know which seven movies I'll see during my winter vacations:

- Hamilton
- Modern Times
- City Lights
- Cinema Paradiso
- Ayla: The Daughter of War
- 3 Idiots
- Dr. Strangelove or: How I Learned to Stop Worrying and Love

Conclusion

Python is an incredibly useful tool with a huge range of open-source libraries for the development of Data Science project, however, without data, it is not that useful for analytics, so that's where datasets enter the game. Relational databases are an extremely efficient, powerful, and widely used way to create, read, update, and delete data of all kinds. The most common relational database management systems are all based on Structured Query Language (SQL); this means that Data Scientists needs a strong understanding on using Python and SQL together to have an advantage when it comes to working with data analytics.

With the completion of this project, we were able to put into practice the knowledge acquired over the last few months. By establishing a statement problem and using tools like SQL and Python we were able to build and analyze a dataset that provided a data-driven solution. Without a doubt, it was a great challenge whose conclusion brings satisfaction, but above all, the desire to continue learning and improving our knowledge in the world of Data Science.