An In-Depth IMDB Review Analysis

SEIS 745 – Data Lake Engineering

Team Members: Pragya Verma and Priyanka Namdev Patil

Introduction: In today's world, when we talk about movies, it's not just about what Hollywood or Bollywood says is good. We're diving into a bunch of movie reviews on IMDB, where people like you and me share what they really think. This collection of reviews isn't just random thoughts; it's like a peek into what a whole bunch of people feel about different movies. The data we're looking at includes stuff like movie names, how much people liked them, when they wrote about it, what exactly they said, and whether they spilled any spoilers.
This project is an attempt to identify necessary information to make better movies, producers can figure out if folks liked what they made, and for us regular movie buffs, it's a space to chat about our favorite flicks. So, as we go through this data, we're basically trying to figure out what everyone's saying about movies on IMDB and what we can learn from it.

Databricks:

- Analyze using Databricks Notebooks on the Unified Analytics Platform, employing Spark for distributed data processing.
- Store data in the Databricks File System (DBFS).
- Code models using Python and Spark (Pyspark) to leverage distributed computing capabilities.
- Employ Python for visualization in Databricks notebooks, utilizing libraries like Matplotlib, Databricks supports various Python visualization tools for data exploration.

IMDB review dataset:
https://www.kaggle.com/datasets/ebiswas/imdb-review-dataset

Kaggle API:

- This data set was in JSON format, we used API based approach to connect the data set.

Allowing us to interact with Kaggle's API from your Python environment.

*All snapshot from Databricks environment is provided in the appendix.
**Code : https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/5423078990868970/1507982150210872/5315594342668074/latest.html

Data Collection: At the initial level data collection starts by installing Kaggle for simplified dataset access getting an app for easier downloads. Credentials are then set up, using API acting as a username and key, ensuring proper authorization to acquire datasets. The actual dataset IMDb reviews, is obtained using Kaggle, and Python tools are then prepared for subsequent data analysis.

Challenges in data collection:

- Since the dataset was in json, we were trying a code which started giving error we then realized it is multiline.
- The data was huge and had multiple files, so it took time for processing. We used union to join these files to create one master file.
- The date format was in d mm yyyy which was giving us hard time to convert it into standard format i.e. mm dd yyyy. So, we decided to split and transform the date.

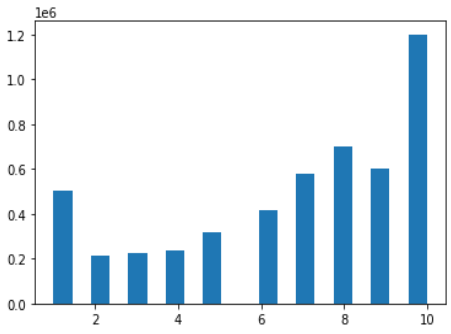
Data Analysis:

The data analysis was done through the pyspark code and python sql libraries. There were multiple levels of analysis.

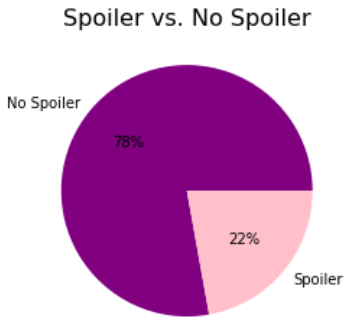
Some of the significant insights after data analysis are as follows:

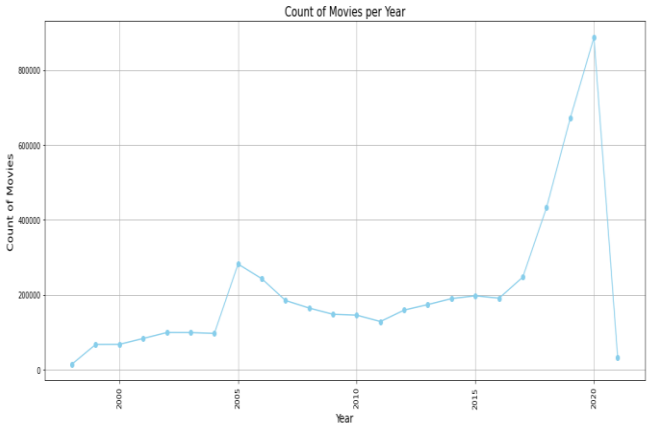
- The dataset encompasses a substantial total of 4,996,558 movies.
- Movie ratings range from 1 to 10, with the highest rating being 10 and the lowest 1.
- The maximum number of reviews given for a single movie is notably high at 7,188.
- A breakdown of reviews indicates that 78% are categorized as non-spoiler, while 22% are labeled as spoiler reviews.
- Analysis of the dataset highlights the top 10 most reviewed movies, providing insights into audience attention.
- Examining the trend of movies produced each year reveals a significant concentration in the year 2020, suggesting a noteworthy production spike during that period.

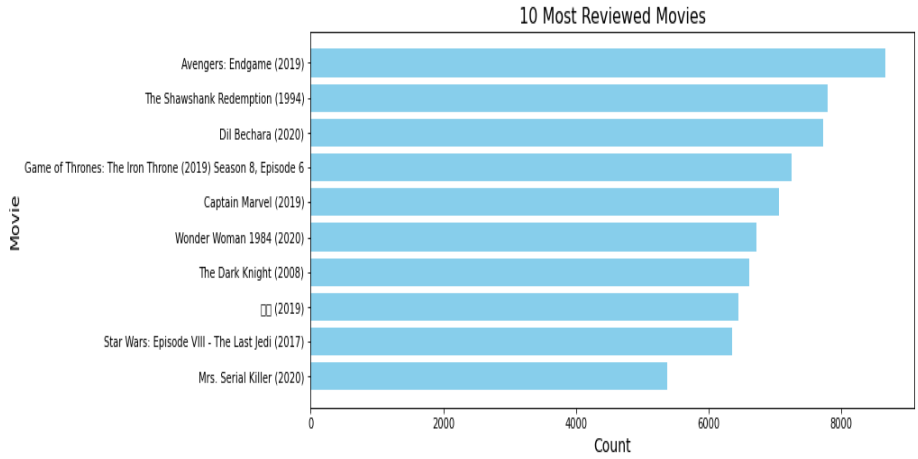
Visualizations:



movie	count
Dil Bechara (2020)	7188
The Shawshank Red...	5392
Avengers: Endgame...	4363
小丑 (2019)	4145
Mrs. Serial Kille...	3978
The Dark Knight (...	3786
The Chosen (2017-)	2925
The Lord of the R...	2898
Scam 1992: The Ha...	2764
The Godfather (1972)	2590







References:

https://www.kaggle.com/datasets/ebiswas/imdb-review-dataset
https://spark.apache.org/docs/latest/api/python/index.html
https://stackoverflow.com/questions/30949202/spark-dataframe-timestamptype-how-to-get-year-month-day-values-from-field