# MovieLens Analysis: Between Two Decades

# Project Proposal

## Background

MoviesLens Analysis: Between Two Decades is a big-data analysis project focusing on the movies over the past 20 years, based on the rating and tags from the movie database site MovieLens.org, to find out how the how the movie production industry, as well as audiences' evaluations and tastes in movies have changed between the decades 2002-2011 and 2012-2021.

The dataset I used was *MovieLens Tag Genome Dataset 2021* from the grouplens website. It contains data on 84,661 movies, 1,094 tags, 212,704 tag-movie relations, and 28.5 million ratings. The last update of the dataset was in 2021, so the analysis goes through the two decades mentioned above.

## Importance

The Hollywood film industry is currently entering a very difficult period, with a large number of A-list productions flopping at the box office and strikes by the WGA and SAG still going on. Many in the industry are also criticizing the fact that movies are gradually losing creativity. Therefore, it is necessary for Hollywood to find out the changes in the audience's tastes and evaluate new projects, based on big data analysis.

The movie industry entered a new era with the release of *The Avengers* in 2012, when studios began to develop the concept of 'cinematic universe'. This analysis happens to use 2012 as the demarcation point, which can effectively reflect the changes in the movie industry and audience between the two phases.

## Features

This project uses applications Jupyter Notebook, Python and Spark SQL to perform the analyzation process using a big scale of data on cloud. The results are visualized to demonstrate how the rating distribution changes, and which types of movies are among most popular and high-rating between two decades. The trends and changes of two period are also included in the analyzation and visualization.

## Traditional Limitation

Traditional computing suffers from low-performance and low-interactivity. Single-purposed applications aren't very helpful if I want to use multiple functionalities from different platforms. If hardware failure happens, it's not easy to recover all the data.

## Cloud Benefits

With cloud computing, I can group various nodes in a cluster, utilize multiple useful tools in one unified stack, easy to interact and analyze with all the functionalities. I can deal with queries and visualization in single workflow without worrying about transfering and compatibility issues.

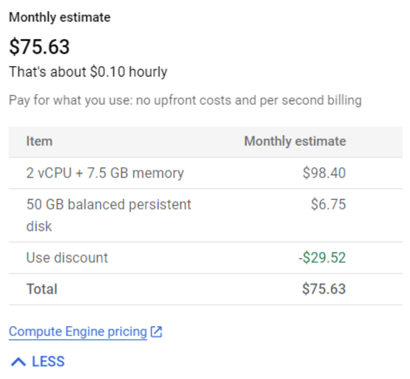
Also, the data process speed is also a big improvement from traditional computing. Higher processing speed and simplified workflow means I can spend less time to achieve this big data analytic project. Last, the recovery and backup on cloud can prevent potential file and data loss issues.

## Cloud Technologies

In this data analysis project, I used these cloud technologies:

- Apache Spark: An open-source distributed general-purpose cluster-computing framework. It works as a unified analytics engine for large-scale data processing, has multiple useful libraries. Fast speed in-memory computing.
- Jupyter Notebook: A web-based interactive computational environment for creating Jupyter notebook documents. It is a high-interactive coding tool, supporting several programming languages and file types, and code can run in cells for maximum convenience.
- Hadoop Distributed File System: A distributed file system designed to run on commodity hardware. Provides high throughput access to application data and is suitable for applications that have large data sets. Also can be utilized to interact with different clients for different uses.

## Monthly Cost and Framework Architecture


Monthly estimate

$75.63

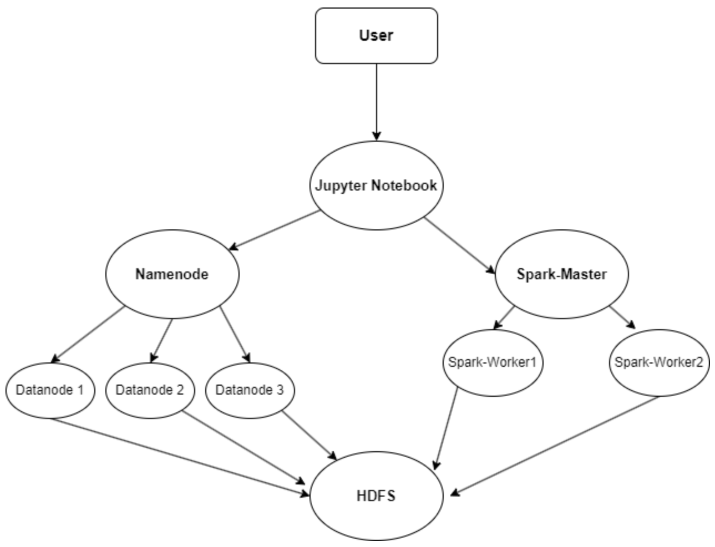
That's about $0.10 hourly

Pay for what you use: no upfront costs and per second billing

| Item | Monthly estimate |
|---|---|
| 2 vCPU + 7.5 GB memory | $98.40 |
| 50 GB balanced persistent disk | $6.75 |
| Use discount | -$29.52 |
| Total | $75.63 |

Compute Engine pricing

LESS



## References

[1] Overview - Spark 3.5.0 Documentation https://spark.apache.org/docs/latest/

[2] Jupyter Notebook, https://en.wikipedia.org/wiki/Project_Jupyter

[3] HDFS Architecture Guide, https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html