**Introduction:**

Recommendation engine are turned out to be a business in current OTT platform. Many Online applications like Netflix, Amazon prime video, Disney+Hotstar uses Recommendation engine in their own way to recommend movies, shows to their users. In parallel, users expect the betterment in such application to get the similar content which saves their money and time.

There are two types of recommendation engine: Content based and Collaborative filtering. Content-based filtering methods are based on a description of the item and a profile of the user's preferences and meanwhile, Collaborative filtering is based on the assumption that people who agreed in the past will agree in the future, and that they will like similar kinds of items as they liked in the past.

Our aim here to create a web application for content-based recommendation engine by taking publicly available movie dataset which contains 500,000 different movies. In this project, we will aim to clean the dataset using Apache PIG and application will be built using Flask python by hosting it in Heroku PaaS Cloud platform.

**Technologies Used**

1. Cloud Platforms:

* **Google Cloud Platform (GCP):** Clusters are created and uploaded the dataset onto the remote server for the further processing.
* **Heroku:** It is a Cloud platform (PaaS) which provides infrastructure to build, host and monitor the application in cloud.

1. Pre-processing the dataset

* **Apache Hadoop:** Basic need of Hadoop was used to store the data
* **Apache Pig:** Pig was used to load and clean the dataset, to deal with missing values, Special characters and unused column which is then pushed to HDFS to do the further processing.
* **Apache Hive:** we used to create and query our database residing in Hadoop and built a visualization table in Tableau by referring it.

1. Building Model

* **PyCharm:** PyCharm JetBrains is made used to build the model by using Python Programming language and also to create the front-end of the web page using HTML and CSS.
* **GitHub**: All files are uploaded to Github in order to give access to the Heroku cloud to deploy the application.

1. Screen Cast:

* **Zoom:** This application was used to create screen cast for the project.

1. Communication

* **GoogleMeet:** This software was used to communicate among the teammates.

**Data Source:**

The dataset is freely available in Kaggle and Github as well. Basically, we combined four different CSV files into one in order have more movie details. The size of the dataset is 500,000 movie details.

The file contains 26 columns, and for the purpose of content-based recommendation engine system we have considered Title, Vote\_Count and Genres. In the application, the three column details will be updated based on the user selected movie.

**Data Pre-Processing:**

**Dataproc and VM-Instances:**

Initially, a master node and 2 worker nodes were created and the master instance was started which is relatively easier way to use cloud service for running onto Hadoop environment.

**ETL Process:**

The dataset was in four different CSV files, so we initially combined the four CSV files. There was a special character in the genres columns, so the column was cleaned. There are total of 26 columns are there in the dataset, and to build the application we do not require whole of them. So, we have considered only Title, Vote\_count and Genres columns and the remaining columns are unconsidered. And later on, the csv files are loaded and stored in HDFS. The whole extract, load and transfer of files are done using Apache PIG.

**Hive Query:**

The processed and cleaned dataset was selected by Apache Hive. A number of queries were written in HQL and the output of these were used within our Tableau Desktop for visualization.

**Building the Model:**

A python web-framework called “Flask” was used to build the content-based recommendation engine application using cosine similarity to measure the similarity between two vectors to point out whether it is in the same direction. So that, based on user likable choices of movies will be recommended for the user when he/she searches for a movie., i.e., if user like to watch a movie “The Avengers”, then similar movies will be displayed based on genre and also displays vote\_count which helps user to choose the next movie he /she wants to watch for.

To do the processing through cloud, Firstly, we dumped all files into github including python and html files. We created a free account in Heroku app and we generated an application by sinking our github repository. A Procfile is included into the github, which include the commands that Heroku app should execute to deploy the application. Whenever the push is done to github files, automatically it updates the Heroku app which is deployed in cloud. When the application is opened, it will ask for the user to enter a movie and when user enters the movie name, similar movie choices will be displayed by redirecting to the next webpage in application by displaying the movie name, vote count and genres of top 10 movies and also a special setting button is included which displays the bar chart of the top ratings movies in the database. If there is case where, user entered which is out of database, then an error page will be loaded which displayes, the movie name doesn’t exist.

**Data Visualization:**

In order to visualize our data, the Tableau was used by comparing the rating and user table based on their age group. Then, to explore the data, the bar chart is displayed by comparing top rated movies and popular genres. And then, content-based filtering is visualized by using Root mean squared error technique.

In the hosted application, the link for the Tableau public desktop is provided, so that the data analysis on the dataset can be found which will be helpful to under the algorithm which is used to build the application hosted in Heroku app cloud platform.

**Challenges:**

Within the give time, we have constructed an application which recommend movies based on user preferences of watching and we are satisfied with our approach.

However, we have also faced few difficulties and challenges while constructing the application. The Google cloud platform was unable to load the Flask server and we tried multiple times. So, we opted for another free cloud PaaS platform Heroku where running a web app is pretty much easier and can be done in minutes.

Furthermore, we tried connecting the tableau with Flask application, but the flask application did not support the tableau application onto the web. So, we included the link which redirects to the tableau desktop page and gives the analysis of the entire dataset.

For our future work, we will continue to work in this project to build a recommendation engine using collaborative filtering. And also, we will explore google cloud and Flask to connect for the tableau application will help us to comminate between the clusters and the tableau desktop tool.

**Responsibility statement**

Below summary illustrates the contribution of each of the team members throughout the project:

1. Sharath Madhu Manjunath – Satisfactory

* Idea of Application (11-NOV-2020)
* Design of Algorithm and application (11-NOV-2020)
* Flask Application in python (PyCharm) (17-NOV-2020)
* Setting up an account and deploying it in Heroku cloud platform (06-DEC-2020)
* Visualization in Tableau (12-DEC-2020)
* Updating the Github to give access to the repository for the Heroku application. (06-DEC-2020)
* Setting up a screen cast video which is uploaded to youtube. (18-DEC-2020)

1. Manju Gurjar – Satisfactory

* Establishing the cluster (12-NOV-2020)
* Pre-processing and cleaning of data in PIG and python (12-NOV-2020)
* ETL process (12-NOV-2020)
* Primary person in developing the web pages (19-NOV-2020)
* Tried to add Tableau to cluster, but was unsuccessful to do it. (21-NOV-2020)
* Involved in designing the application. (21-NOV-2020)

1. Dusha Saravanan – Satisfactory

* Data Analysis (11-NOV-2020)
* Data Visualization in Tableau (12-DEC-2020)
* Setting up a Tableau public desktop (19-NOV-2020)
* Updating the project report (17-DEC-2020)
* Initially tried to run flask program in Anaconda Jupyter Notebooks. (12-NOV-2020)
* Tried to implement collaborative filtering technique of recommendation (15-DEC-2020)
* Secondary person in developing the web pages with Manju Gurjar (21-NOV-2020)

**Github:** <https://github.com/sharath30/Movie-Recommendation-Engine>