|  |
| --- |
| TECHNICAL REPORT |

|  |
| --- |
| Electrical & Computer Engineering & Computer Science (ECECS) |

|  |  |
| --- | --- |
| SPRING 22 |  |



Contents

[Project Name 2](#_Toc96341549)

[Executive Summary 2](#_Toc96341550)

[Technical Report 3](#_Toc96341551)

[Highlights of Project 3](file:///C:\Users\ardia\Desktop\TEACHING\SPRING%2022\DSCI6002%20Data%20Science\Projects\Technical%20report%20Template.docx#_Toc96341552)

[Submitted on: 3](file:///C:\Users\ardia\Desktop\TEACHING\SPRING%2022\DSCI6002%20Data%20Science\Projects\Technical%20report%20Template.docx#_Toc96341553)

[Abstract 4](#_Toc96341554)

[Methodology 5](#_Toc96341555)

[Results Section 7](#_Toc96341556)

[Discussion 8](#_Toc96341557)

[Conclusion 8](#_Toc96341558)

[Contributions/References 9](#_Toc96341559)

|  |
| --- |
| Netflix Movie Recommendation |

|  |  |  |
| --- | --- | --- |
| Executive Summary In this project, we will be predicting the Netflix movie recommendation using Spark on Amazon EMR cluster. The dataset we are using here is a subset of Netflix price data released by Netflix. It contains various information on 100000 ratings users posted on around 17 thousand movies. Using this dataset, we will be predicting the top 25 best movies which are to be appeared on Netflix and compare it to the actual recommendation on the Netflix. We will be using various methods from the os and pyspark as the project is completely based on python. | | |
| person at a table writing in a notebook with people around | | |
| **Team Members:**  **Rakesh Reddy Poddutoori**  **Nakul Reddy Nimmala**  **Navya Gorantla** | **Questions?**  Contact: Rakesh Reddy Poddutoori |  |

|  |
| --- |
| Technical Report |

**Netflix Movie Recommendation**



## Highlights of Project

* Using Netflix’s own data making a similar movie recommendation algorithm and comparing it to actual Netflix recommendation.
* Exploring the various methods from pyspark whether it be SQL related or ML related pyspark provides us with many sub-methods which are useful.
* Making the use of AWS S3 buckets, EMR clusters and completely carrying out the process on it using Jupyter notebook internally. Here, all processes are carried away on the remote AWS server rather than traditional local-host.
* Carrying out the prediction using algorithms like Collaborative Filtering, Alternating least squares for machine learning and resulting regression is evaluated by RegressionEvaluator() and getting useful insights from it.

## Submitted on: MAY 3,2022

## Abstract

This project is about creating predictions for top Netflix movie recommendations It involves in utilizing Spark on an Amazon EMR cluster to forecast Netflix movie recommendations. The entire project is based on python and methods involved in it on AWS EMR cluster. The dataset we're working with is a subset of Netflix's prize data. It contains data from over 100,000 user-submitted ratings on over 17,000 films. We will use this dataset to forecast the top 25 movies based on ratings that will be available on Netflix and compare them to the real Netflix recommendations. Various machine learning methods are utilized in this project and eventually carry away the operations.

Notebook Pitch: <https://colab.research.google.com/drive/1iMu4wLwmSUR8zkF6paDvDj-MqnF0QliP?usp=sharing>

Cover Page

Title of the Report: Netflix Movie Recommendation

Authors: Rakesh Reddy Poddutoori, Nakul Reddy Nimmala, Navya Gorantla

Contact: [nnimm2@unh.newhaven.edu](mailto:nnimm2@unh.newhaven.edu)

Date of Publication: May 3,2022

Introductory Section

Ever wonder how Netflix recommendation algorithm works? On what basis the Netflix recommends us good movies which are good to watch and makes us believe in it while watching those recommendations? This project is all about create one such recommendation algorithm completely on Jupyter Notebook on AWS server.

Review of available research

The available research contains the usage of only machine learning algorithms based on very limited parameters. The resulting model is very less affective when compared to our model. The currently available methods for recommendation also lack the remote usage of the model and the correct implementation of pyspark. Available models remain very far behind when predicting the Netflix recommendation and the accuracy scores remain very low. However, in this project we overcome all such possibilities and create a very effective prediction algorithm for Netflix recommendation.

## 

## Methodology

In this project, we will be initializing the S3 bucket and store our data set in it. Then create the EMR cluster and SSH into it using PuTTy. After that create a virtual environment for the jupyter notebook and continuing the process ahead.

We first import the libraries of os unittest and spark which contain useful methods to carryout our processes.

Then we load and cache our dataset from S3 bucket by setting the directory.

We then do the exploratory data analysis and find insights on dataset as we find out that there are 100477 ratings on 17769 movies in the dataset.

Basic Recommendations are first set based on average ratings of the movies, their names and their ids. This step involves in sql methods from pyspark.

Then movies with highest average ratings and atleast with 500 reviews are segregated later on to apply collaborative filtering.

Here, we first split the data set into three parts using randomSplit() from pyspark at ratios of : 0.6 ( for training) , 0.2(for validation) and 0.2 ( for testing) and visualize each at 60354,20041 and 20082 entries respectively.

We then apply Alternative Least Square(ALS) learner from pyspark.ml.recommendation for dataset and then make the predictive model and evaluate their rank

Then we test the model, and compare RMSE of the model’s projected results to the values in the test set.

Later on, we make predictions for ourself by importing pyspark’s sql method ‘Row’ and segregating information in the way of one user id and movies watched by that person. Then adding more movies to this sub-set and training the model in similar way as previously. Followed by checking RSME to the new model with our ratings.

Since till now we calculated error in the model, we now proceed for prediction and avoid losses by giving rating for unrated ones. Later on predicting the ratings for the same dataset.

Now that we have seen ratings, then rated, predicted and checked the ratings with each other. We get the final recommendations based on the ratings and preferences provided.

We view and compare them to the actual Netflix recommendation and view our efficiency.

We hence created the top 25 recommendation of movies from Netflix similar to its’s own predictive algorithm.

## 

## Results Section

## 

We can see the top 25 movie recommendation as the results of our implementation of the model.

## 

## Discussion

When we discuss the results we can clearly find out that based on the simple mechanisms on AWS EMR cluster we can create an effective prediction model for movie recommendation on Netflix.

As the results page suggests the movies more than 75 reviews are taken into consideration as the best fit recommendation only happens with the ones with more approval ratings.

Implementation of collaborative filtering and machine learning methods of Alternative least squares we had created our model and compared RMSE to make it more predictively efficient.

Results table contain column parameters of average ratings, title, movie\_id ,count and prediction metrics(through which we can determine efficiency).

This model is has got good predictive power and was able to make top 25 movie recommendation efficiently with minimal error.

## Conclusion

Hence we can conclude that using Spark on EMR cluster we executed the program for rating prediction and evaluated the recommendations with provided true Netflix recommendation and thus validating our program efficiency. All this work is made in python execution environment of Jupyter.

## 

## Contributions/References

1. <http://www.netflixprize.com/>

2. <https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data>

3. <https://towardsdatascience.com/the-netflix-prize-how-even-ai-leaders-can-trip-up-5c1f38e95c9f>