CS 6240: Project Proposal

AMAZON MOVIE REVIEWS ANALYSIS

Team Members:

Name	**Email ID**	**Section**
Pavitra Srinivasan	srinivasan.p@husky.neu.edu	01
Nasir Ahmed Raffick	raffick.n@husky.neu.edu	02
Siddarthan Visvanathan	visvanathan.s@husky.neu.edu	02
Dev Pranav Puchakayala	puchakayala.d@husky.neu.edu	02

Dataset:

We are working with the Amazon Movie Review dataset. We obtained the dataset from the following link: https://snap.stanford.edu/data/web-Movies.html

This dataset consists of movie reviews from Amazon. The data span a period of more than 10 years, including all ~8 million reviews up to October 2012. Reviews include product and user information, ratings, and a plaintext review. Following statistics has been provided for this dataset.

Dataset statistics	
Number of reviews	7,911,684
Number of users	889,176
Number of products	253,059
Users with > 50 reviews	16,341
Median no. of words per review	101
Timespan	Aug 1997 - Oct 2012

File	Description	Size
movies.txt.gz	Amazon movie data (~8 million reviews)	9.3GB

Dataset Format:

The Amazon movie review dataset is provided in **movies.txt.gz file** in the following text format:

```
product/productId: B00006HAXW
review/userId: A1RSDE90N6RSZF
review/profileName: Joseph M. Kotow
review/helpfulness: 9/9
review/score: 5.0
review/time: 1042502400
review/summary: Pittsburgh - Home of the OLDIES
review/text: I have all of the doo wop DVD's and this one is as good or better than the 1st ones.
```

where

- product/productId: id of the product which can be accessed as amazon.com/dp/B00006HAXW
- review/userId: id of the user
- review/profileName: name of the user
- review/helpfulness: fraction of users who found the review helpful

- review/score: rating of the product
- review/time: time of the review (unix time)
- review/summary: review summary
- review/text: text of the review

Major Tasks:

1. Creating a Web Interface and creating cluster using the AWS CREATE CLUSTER SDK

The purpose of this task is to create a single user web interface, through which a user can select the required tasks to be run, which are specified below. The main advantage of this task is that the cluster is created programmatically, thus allowing us to dynamically use the user's input to run the corresponding Map Reduce Task.
The link for AWS SDK to create cluster:
http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/calling-emr-with-java-sdk.html

2. Top 5 useful reviewers:

In this task, we are planning to analyze the "review/helpfulness" field from the dataset and determine the top 5 reviewers whose reviews were considered as helpful by other users based on the rating provided by the users to the reviewer. We may have same user review multiple movies/video. So we will aggregate those results to find the top 5 reviewers whose reviews have been helpful for motivating other users to buy that movie/video.

3. Data Feature Enhancement

In this task, we are going to use the "product/productId" field to first identify the product details by passing the ID to the Amazon Product Advertising API. So the product details would be given in the following format:

```
<Item>
        <ASIN>B000A2XB9U</ASIN>
        <ItemAttributes>
                <Director> James Cameron </Director>
                <EAN>0014381273229</EAN>
                <Format>Color</Format>
                <Language>
                        <Name>English</Name>
                        <Type>Original Language</Type>
                </Language>
                <ListPrice>
                        <CurrencyCode>USD</CurrencyCode>
                        <FormattedPrice>$19.99</FormattedPrice>
                </ListPrice>
                <NumberOfItems>1</NumberOfItems>
                <ProductGroup>DVD</ProductGroup>
                <ReleaseDate>2009-12-18-</ReleaseDate>
                <Studio>Image Entertainment</Studio>
                <Title>Avatar</Title>
        </ItemAttributes>
</Item>
```

Then by passing the <Title> and <ReleaseDate> field to the OMDb API we can get the information related to the movie. The data returned is in the following format:

```
{
"Title": "Avatar",
"Year": "2009",
"Rated": "PG-13",
"Released": "18 Dec 2009",
"Runtime": "162 min",
"Genre": "Action, Adventure, Fantasy",
"Director": "James Cameron",
"Writer": "James Cameron",
"Actors": "Sam Worthington, Zoe Saldana, Sigourney Weaver, Stephen Lang",
"Plot": "When his brother is killed in a robbery, paraplegic Marine Jake Sully…",
"Language": "English, Spanish",
"Country": "USA, UK",
"Awards": "Won 3 Oscars. Another 84 wins & 106 nominations.",
"Poster": "http://ia.media-imdb.com/images/M/Ml5BMl5BanBnXkFTcwODc5MTUwMw._V1_SX300.jpg",
"Metascore": "83",
"imdbRating": "7.9",
"imdbVotes": "818,467",
"imdbID": "tt0499549",
"Type": "movie",
"Response": "True"
}
```

The above data is merged with the original dataset for all matching records found in OMDb, thus enhancing the features of movie records, which eventually increases the dataset size. This new information obtained can be used for further analysis tasks.

We will be using Apache SparkSQL database to store this equi-joined Data with all the required attributes.

4. Recommending Movies

This task makes use of the above mentioned web interface, through which user can select options to see top 5 movies in each Genre that:

- won Oscars and other awards.
- has high IMDB ratings.

The data is served from SparkSQL database, which allows faster response for a given query.