**Final Project**

**Team -02**

**Dwiz Kumar**

**Prajakta Deshpande**

**Purva Redkar**

**Salman Khan**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| Sr. No | Title | Page Number |
| 1. | Core Algorithm | 3 |
| 2. | Backend | 3 |
| 3. | Data | 3 |
| 4. | Business Logic | 4 |
| 5. | User Interface | 5 |
| 6. | UI Snapshots | 5 |
| 7. | Test Results | 8 |
| 8. | Set Up | 9 |
| 9. | References | 9 |

In the given checkpoint we have implemented core algorithm from scratch using java. Previously, we have implemented it in python using the library in the chekpoint3.

In addition to the core algorithm, we have also implemented user interface using android application.

**Core Algorithm:**

**Sentimental Analysis:**

We have implemented Naïve Bayes algorithm for sentimental analysis. In Naïve Bayes, we have used two different concepts – conditional probability, and Laplace smoothing. As a part of this implementation we have trained users’ comments and categorize it as positive and negative review classes. Once the application is trained, based on training set, application will categorize a new set of record and will classify it as positive/negative review.

**Recommendationation:**

Our team has implemented Cosine similarity to display similar movie name. As a part of this implementation, we first calculate term frequency and document frequency. Using these two terms we calculate tf and idf. Followed by this we calculate weight and then cosine similarity. As a part of result, if a user inserts a movie name in the search option, they are expected to get similar movie name as a part of recommendation. If a movie name is not present user will get “no relate movie found” message.

**Backends:**

For the backend our team has initially (for checkpoint 3) implemented Naïve Bayes and recommender algorithm in python. Our initial decision to choose python was two folds. Firstly, python comes with in-built libraries for machine learning and text- mining. For example, we have used inbuilt library named pandas for machine learning, which provides machine learning functionalities, one other library is NLTK, which we had used for removing stop words. Secondly, python takes lesser development time. After checkpoint 3 grades, we got to know that we have to implement both the algorithms without using any libraries. So, we decided to move back in Java and we actually used some part of code taught in the class in our implementation and extended it to the functionality that was required.

The complete source code of both the algorithms are provided with the dataset in the deliverables. Moreover, we have included necessary comments and java-docs for clean code and make the code more readable and represent it as a document.

**Data:**

The data is extracted from Amazon Prime. According to a report, as of April 2017, Amazon prime has over 80 million paid users [1] which generate large volume of data. Every video has comments or reviews that are given by users. As amazon prime videos has to be subscribed, it is important to know whether any video is worth watching. These comments are users’ reviews who have already seen the video. We have collected these comments, the comments are linked to a videoID(showid), reviewerid, reviewername, reviewertext, summary, review time. We have collected around 37,126 reviews. Data being the driving factor of the project, comments have taken into consideration for the further processing for the video which have minimum 5 reviews. This makes the data set dense and has no duplicate records. The dataset[2] that our team has taken for sentimental analysis and recommendation are in json format. We have chosen dataset in this format because json format gives a name-value format which makes developers life easy to extract data.

Most widely used words in the comments are highlighted using word cloud. The below screenshot shows the most used words by reviewer in the comments. The word cloud shows positive words such as great, good, love, like and negative words such bad, no tv fun, never, never seen.

****

**Business Logic:**

Our team’s overall goal was to develop an application that will help user to identify sentiments of other users via existing comments and based on that user will get a small description about rating as well as existing comments’ sentiments. There is another part in the business logic where user will also get some recommended movies name, which probably they will like to watch. Overall, our application (Amazon prime video analysis) will help user to give information about movie and that will help them to take decision that should they watch the movie or not. It also helps them to come out of cumbersome task to reading all the comments about a movie to identify whether they should watch or not. Instead of the that, in just one click user will get short and summarized information.

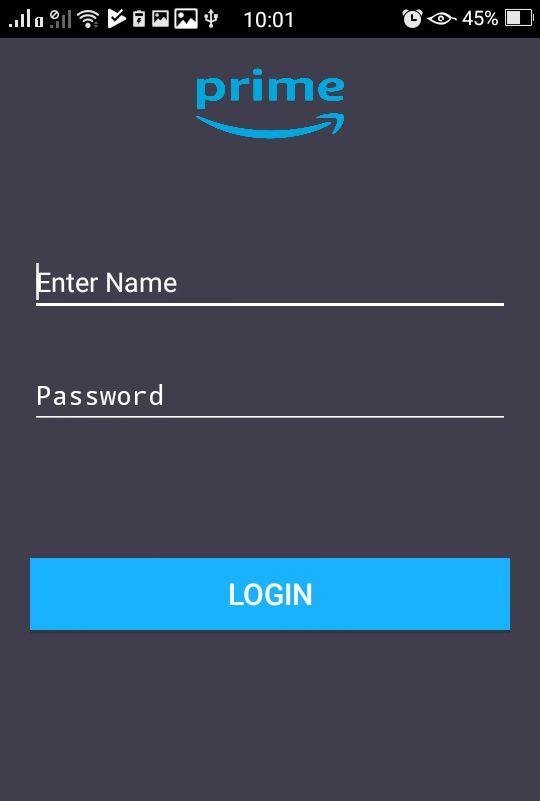
**User Interface(UI):**

For user interface we have implemented android app. UI has two parts- login page and core business functionality page. Once the user opens the app, user will get a login screen page which will take user name and password. We kept this feature since amazon prime is a paid service and it requires user to login. Once user is successfully logged in, then they will be navigated to the business functionality page where there are five boxes.

* **Movie Name:** Takes the movie name as input and search that whether movie name is existing in the data set or not.
* **Display property:** Displays name of movie searched, overall rating of movie, and how the overall review sounds positive/negative.
* **Comment:** Comment takes user’s comment as input and evaluates it that whether comment sounds positive/negative.
* **Recommend result:** This textbox displays list of movie title which user are recommended to watch based on the movie name searched in the movie name text box.

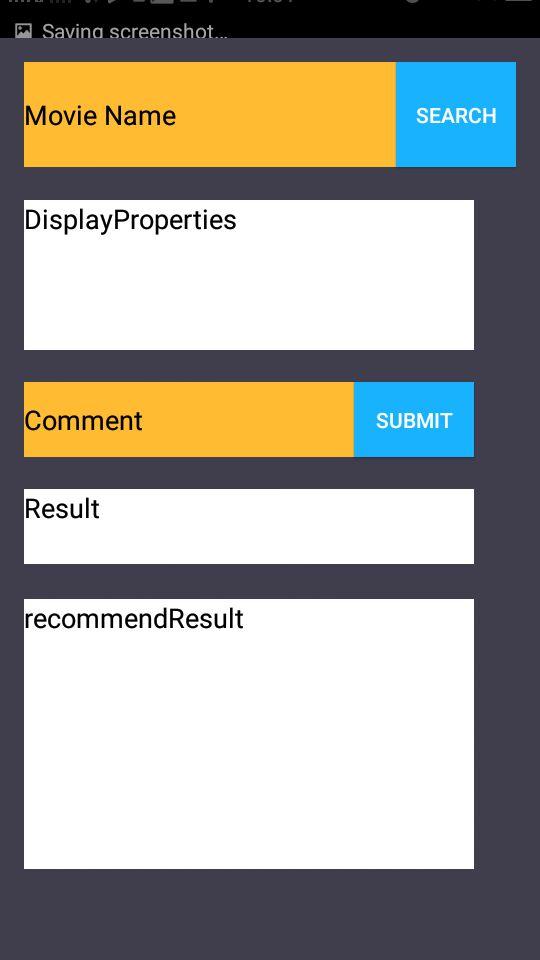
**UI Snapshots:**

1. **User Login Page:**

****

1. **Business functionality page(without user’s input)**

* 1st test box is where name of movie is supposed to enter. Once the user enters a movie name and click on search.
* Search result gets populated in DisplayProperties textbox. Search results gives the name of the movie, ratings, and kind of overall review movie received by our classification implementation (positive/negative)
* Followed by that there is a comment text box where the user can give their own comment. Once the user gives comment and click on submit button.
* User will get a response that how the comment actually sounds positive/negative.
* Last textbox is a recommended result which gives a list of movies that matches with the provide movie name.

****

**C) Business functionality page(with user’s input)**

A screenshot of a cell phone

Description generated with very high confidence

**Test Results:**

We have tested our application for below 6 different test cases:

1. User should get login screen once they open the app.
2. Correctly classifying positive comments
3. Correctly classifying negative comments
4. Recommendation System: Recommendation for 2 similar Movie/tv show. Here when user enters any respective video/tv name the system will give top 2 recommendations
5. Exception handle: When user inserts a movie name which is invalid/not in database.

This is first level of unit testing. We had listed down all possible scenarios based on our understanding and came to a conclusion that these test cases are sufficient for unit testing.

Note: More details are available at TestCases\_Final.xls

**Setup:**

* Download and run the app-debug.apk file in an android phone.
* After downloading the app-debug.apk, store the files - AmazonR.csv, positive.txt and negative.txt in your android’s external SDCard. Store the files in SDCard using the path /storage/sdcard1.
* After storing the files, run the installed application and login into the application with username as ‘admin’ and password as ‘admin’. After this the app can be used.

**Reference:**

**1.**  <http://fortune.com/2017/04/25/amazon-prime-growing-fast/>

**2.** http://jmcauley.ucsd.edu/data/amazon/