Sentiment Analysis-Based Recommendation Engine (SABRE)

Achal Shah (1213294158)
anshah9@asu.edu

Sarvesh Patil (1213353386)
sspati13@asu.edu

Giriraj (1213350721)
ggirira2@asu.edu

Shikhar Sharma (1213091904)
sshar133@asu.edu

Nidhi Dubey (1213031246)
ndubey2@asu.edu

Vishwakumar Doshi (1213322381)
vdoshi1@asu.edu

ABSTRACT

In this project we tried to address the limitation of currently used recommendation systems that usually rely on user’s search history and demographic information for making recommendations. Though useful, we believe that in this era of rapidly growing user centric digital developments, it becomes crucial to incorporate sentiments into the loop to make better decisions and tailor certain experiences. Thus, we implemented a recommendation system that, along with other factors, also considers user’s emotional state of mind into consideration. With use, the system learns user’s preferences under different emotional states that helps cater to their needs.

1. INTRODUCTION

Digital presence takes up large portion of our daily lifestyle nowadays. It has become a trend to use the social media to express ourselves and share our activities updates, pictures, or talk about our experiences from the events we attended or places we visited. As users we also come across many recommendation systems acting under the hood on platforms such as Netflix, Spotify, Amazon etc. Hence in this project, we have attempted to bring these two worlds together.

With the rapid progress made in smart home and highly customized user experience, there is a need for intelligent recommendation systems that takes user’s emotional state as a factor for recommendation. The sentences that people write in their posts or captions and their social media activity can help us analyse user’s current mood such as if the user is happy, sad, angry, excited etc. With the knowledge of the sentiments we will have a better idea what type of media and merchandise suggestions user would like the best.

Working on this idea, we have attempted to implement a recommendation model that would track user's sentiment, current mood or state of mind based on his online presence to fine tune items that gets recommended to user at given time; be it a movie, music or merchandise etc.

2. PROBLEM DESCRIPTION:

Recommendation systems are vital component of today's media and merchandise platforms such as YouTube, Amazon, Spotify etc. Although the underlying working mechanism of these systems are based on algorithm that tracks the user's historic data and choices on that service and based on that provides the similar suggestions. For example, on Netflix, based on show or movie that you have watched, Netflix gives you similar category suggestions. To fine tune these recommendations for user, we can incorporate user's feelings and state of mind in making the decisions. That way the content feed given to user would be more dynamic and relevant.

There are two general methods for recommendation

1. Content Based Filtering: The recommendation engine would recommend items like the ones marked positive by user based on discrete characteristics of the item.
2. Collaborative Filtering: The system uses the actions of user to recommend items.
   a. User based: Items that are recommended to a user are based on an evaluation of items by users of the same neighborhood, with whom he/she shares common preferences.
   b. Item Based: Uses the patterns of users who browsed the same item as the current item to recommend next item.

The first challenge in design of such system is getting user's feelings to feed to our decisions making model. One of the best options for us to extract this information about user is through social media feed of that user. We are using posts from user on variety of social media accounts as our root data to form basis of our model's inputs.

Another challenge is maintaining a separate pool or buckets containing selected suggestion options for model that are associated with respective feeling of user. For example, a selection of 5 movies that will be suggested to user if an emotion such as "sadness" is detected through user's feed. Over the time we also need to tune these buckets to be more suited to user as for a common sentiment, the suggested content in bucket may vary from user to user (movies watched by user1 when he/she is sad could be different than user2 when he/she is sad)

Sentiment Analysis or Emotion AI refers to the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis aims to determine the attitude of a speaker, writer, or other subject with respect to some topic or the overall contextual polarity or emotional reaction to a document, interaction, or event. The attitude may be a judgment or evaluation, affective state (the emotional state of the author or speaker), or the intended emotional communication (the emotional effect intended by the author or interlocutor).

In our case for this project we treat sentiment analysis as a model to predict the attitude of the user for reviews for a product in terms of usefulness / likeliness of the product to the user. The sentiment analysis can be broadly classified into following approaches:

1. Rule-based systems: perform sentiment analysis based on a set of manually crafted rules.
2. Automatic systems: rely on machine learning techniques to learn from data.
3. Hybrid systems: combine both rule based and automatic approaches. [1]

For the Sentiment Analysis we implemented the following Automatic approach based on ML.

3. METHODOLOGY

Movie recommendation System:

Datasets: ISEAR, MovieLens
Models: Naive Bayes, K-Nearest Neighbors

The movie recommendation model is based on two components. In first part we are extracting the sentiment from user's posts. Link to third party source code [2] we have used to achieve this is given is references. This model has been trained on ISEAR dataset developed by the Swiss National Center of Competence in Research that contains emotional statements. We are first reading the social media feed to extract the sentences in post. These sentences serve as an input to this emotion detector model. Based on the keywords in that sentence, model applies Naïve Bayes algorithm to detect the emotion. This emotion can be joy, sadness, shame, guilt, disgust, fear and anger. The selected sentiment from above set is then passed as input to next stage of model.

For selecting relevant movies to be suggested, we have used MovieLens dataset [3]. This dataset describes 5-star rating and free-text tagging activity from MovieLens, a movie recommendation service [1]. It contains 27753444 ratings and 1108997 tag applications across 58098 movies. These data were created by 283228 users between January 09, 1995 and September 26, 2018. This dataset was generated on September 26, 2018. This genome dataset uses selected keywords that serve as a feature. The tag genome is a data structure that contains tag relevance scores for movies. The structure is a dense matrix where each movie in the genome has a value for every tag in the genome. For data cleaning, we are removing incomplete data points. Also, initially we have 1128 tags. To reduce them, we are rejecting those tags for which the mean relevance is less than a certain threshold meaning only tags with significant impact in decision for a movie are kept. This greatly reduces the time for finding relevant movies. Using the feature vector of input movie, we implement K-nearest neighbors' algorithm over the dataset to find the 3 closely related movies to our input movie.

For mapping between extracted sentiment and movie recommendation, we have associated a separate bucket for each of the sentiment. In this stage, initially all buckets are initialized with 3 default movie recommendations for each emotion in our set. After getting emotion from the sentiment analysis model, 3 relevant movies are suggested like each movie in the bucket corresponding to that sentiment i.e. 9 movies in total. User has the option of selecting any one of these 9 movies that are suggested for that input sentiment's bucket or may choose any other from the database. The movies in bucket gets replaced by the movie selected by user hence the next time he/she will be suggested based on movies actually watched by them in that emotional state. Thus, over time the model gets tuned to user's preference.

The replacement also takes care that if the movie is watched again, it prevents duplication in the bucket. Also, the deletion takes place in an order which is maintained in a list of indices keeping a track about which movies to be deleted the next time so that the list stays updated.

Music recommendation System:

In our music recommendation model, we are trying to make recommendations which are more personalized for the user. To recommend music we have used both user based and item based collaborative recommendation filtering.

DataSet: Last.fm dataset [4]
Models: Kmeans, K-Nearest Neighbours

To implement our model, we have used the Last.fm dataset which has 360K users from Universitat Pompeu Fabra at Barcelona. The dataset has two parts i.e. two tab separated (tsv) files containing activity data and user data. The activity file has user id, artist music id, artist name and number of plays by a user for an artist. This file details how many times a Last.fm user played songs by various artists. The dataset has 359,347 unique users and total size of activity dataset is 2000000.

Variable Name	Data Type	Sample Data
User Id	String	00000c289a1829a808ac09c00daf10bc3c4e223b
Artist Music Id	String	3bd73256-3905-4f3a-97e2-8b341527f805
Artist Name	String	betty blowtorch
Plays	Int	2137

Table 1: Sample data values in the artist dataset

The other file has user id, gender, age, country and signup date on the last.fm site. This is also a tsv file which will give more personal information about the users. Total size of this dataset is 359347. To join activity dataset and user dataset, we have used User Id as the common key.

Variable Name	Data Type	Sample Data
User Id	String	000063d3fe1cf2ba248b9e3c3f0334845a27a6bf
Gender	String	m
Age	Integer	19.0
Country	String	Mexico
Signup	DataTime	Apr 28, 2008

Table 2: Sample data values in the user dataset

Data Cleaning: For several entries, the dataset had blank spaces or missing values for example in gender or age of the user hence we assigned gender as ‘m’ in those missing cases and assigned mean of ages of all the users as the missing age for a user. After cleaning up the data and we have deleted the user sign up date column as we did not require it for our recommendation model. We used pandas to read the data to build our models.

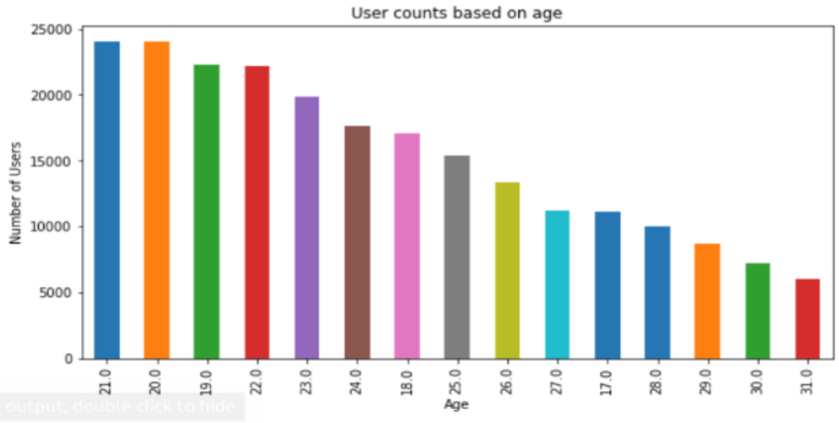


Figure 1: Dataset distribution on basis of user’s age

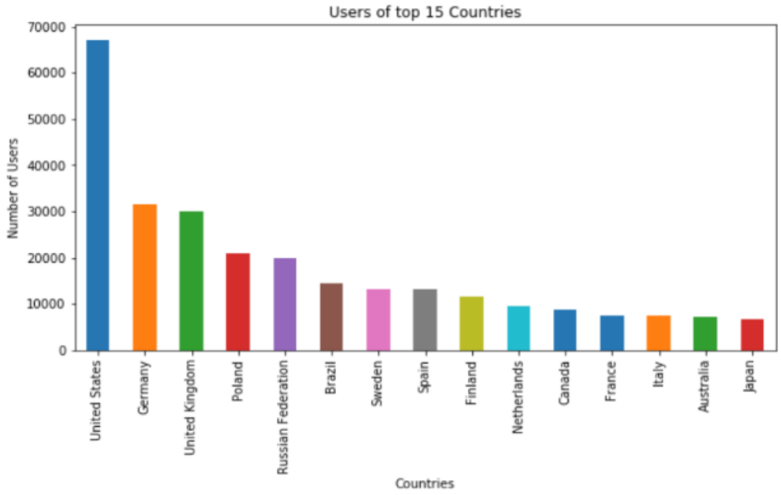


Figure 2: Dataset distribution on basis of user’s country

For item-based filtering we designed a model where we would choose a popular artist at random and suggest 5 artists like this artist to the user. To implement this, we calculated the total number of plays for an artist for all 360K unique users and then added this to the original dataset. We got a final matrix with artists as the rows and users as columns and the number of plays is the value corresponding to each cell in the matrix. We have implemented K-means algorithm based on number of users who played a particular artist song. The model generated a k-nearest neighbor and in this case, we are taking k = 5. To recommend similar artist, we take the k neighbor artists for a given input and display them to the users. Further, to avoid noise we restricted our model to most popular artists for our dataset hence we looked at the top quantile of our dataset.

As we said that we wanted more personalized recommendation, we also added a filter to recommend artists from a particular country to a user. However, we have decided to keep this feature is optional to suit user’s choice. Also, our dataset should have an artist appear just once for a user, so we have removed all duplicates. After this we predicted the top 5 similar artist using K-Nearest-Neighbor approach with k=5. We selected the auto algorithm so that the model would automatically select the best algorithm. We created a matrix that will have artists as rows, users as columns and the total number of plays will be our data value in this matrix. Also, there will be cases when there are no plays by a user for an artist, so we have filled those entries with 0. We created a sparse matrix of this matrix and fit it into our KNN model and calculated distance using cosine method. After this we made recommendations as we selected an artist and model will suggest me 5 artist which have the shortest distance or minimum variation from the artist suggesting that a user who has liked our randomly selected artist might also like these 5 recommendations. For this we have taken some ideas from a similar approach used for recommending music [5].

Above approach is good for the dataset which has the user listening history but when there is no history for the user, we used collaborative approach [6]. In this approach, the model is build using the user's dataset which contains information about users who listened songs for any artist. Model can make some prediction about user's music preference using age, gender and country. To build this model, age, gender and country are mapped with the numerical unique values so that it would be easy for applying ML algorithms. We applied K-Means on this dataset with input features as age, gender and country with K=10. Assign each datapoint to corresponding cluster and map this user data with the user ID. In the merged dataset find the number of top 10 artists in every cluster. To recommend a user, just take input as age, gender and country and the model outputs the recommended artists corresponding to that cluster. While cross validating on many data points, we found the accuracy is good when we this model where the listening data for any user is less comparatively and the other item-based model works well when the users listening data is large.

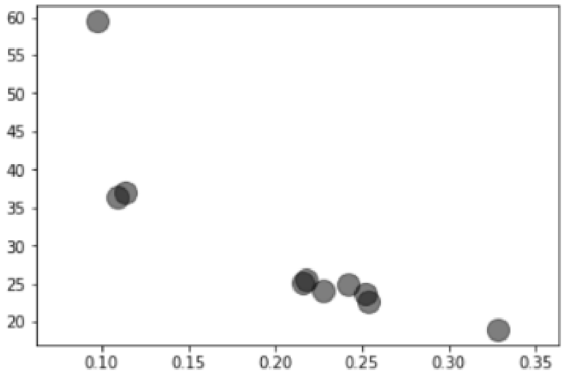


Figure 3: K-means clustering result

Merchandise Recommendations

Dataset: We took the amazon review dataset for Clothes, shoes and jewelleries and beauty products. We considered the reviews and ratings given by the user to different products as well as his/her reviews about his/her experience with the product(s). [7]

Models:

<u>Sentiment analysis</u>:

1. Logistic Regression
2. Naïve Bayes – Multinomial
3. Naïve Bayes – Bernoulli

Recommendation system:

1. k-Nearest Neighbours

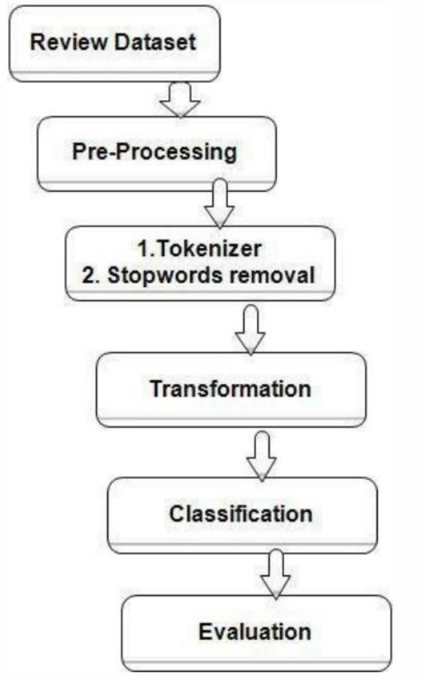


Figure 4: Steps for text analysis

1. Review Dataset: We used the dataset from the amazon which has around 300k reviews for Clothes, Jewellery, Beauty, Shoes products
2. Pre-Processing: This step is to clean the data. Since the review data is in the format containing special characters, etc. we need to remove the and convert the dataset to a clean summary of the review.
3. Tokenizer/Stopwords removal: Here we use the CountVectorizer model to first convert the review dataset into vectors of useful words / phrases in the review. We also use STOPWORDS from python's nltk library to help remove the unnecessary words / characters from the data.
4. Transformation: We used the Tfidf Transformer to convert the tokenized words from the above step to form the weighted tf-idf vectors. Term frequency–inverse document frequency model was applied for the purpose of finding strongly related words for relevant documents.
5. Classification: This is the most important step of the sentiment analysis where we train an ML classifier which can further be used to predict sentiment of any given review. In this step we used 3 different classifiers, Naïve Bayes – Bernoulli, Naïve Bayes – Multinomial and Logistic regression. The input to these models are the tf-idf vectors formed in the above step. And the output is a positive or a negative sentiment.
6. Evaluation: For the evaluation of the above 3 models we divided the dataset into 80% training and 20% testing. The accuracy scores were 89% for NB – Bernoulli, 91% for NB – Multinomial, 94% for Logistic Regression. The figure below shows the comparison of the three models in the terms of the true positive rate vs the false positive rates.

   We can clearly infer from Figure 5, the True positive rate is much higher (with AUC = 0.84) for Logistic Regression model, when compared to other models.

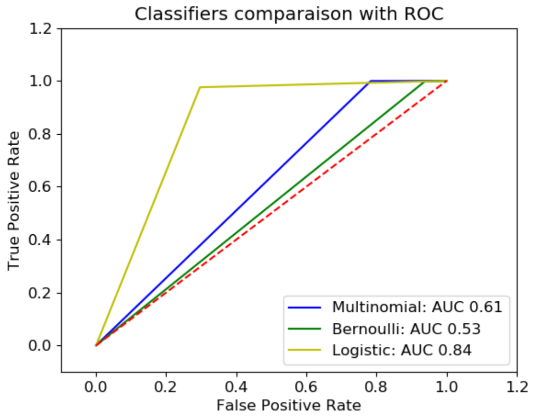


Figure 5: ROC chart

	Precision	Recall	F1- score	support
Positive	0.78	0.70	0.74	5291
Negative	0.97	0.98	0.97	44360
Total/avg	0.94	0.95	0.95	49651

Table 3: Tables showing results for Logistic Regression

Thus, from all the results shown above we can infer that Logistic regression is the best model for sentiment analysis using the automatic approach and on our review dataset. These are only the results for review dataset, however the sentiment analysis using Logistic Regression may perform differently using the other datasets (including conversational sentences).

Now, we will see how these sentiment analysis results help in building a recommender system. Recommender systems for merchandise are an important part of the information and e-commerce ecosystem. They represent a powerful method for enabling users to filter through large information and product spaces. A recommender system/engine is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item[7]. Recommender systems are utilized in a variety of areas including movies, music, news, books, research articles, search queries, social tags, and products in general. In our case we are going to use the recommender system for the products[8].

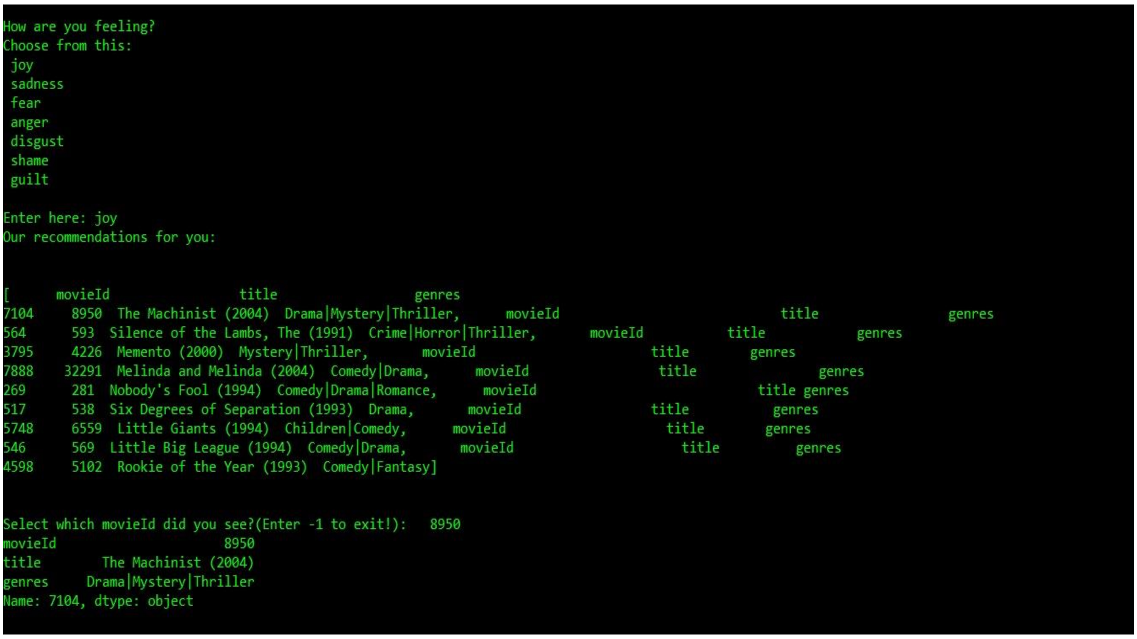
The user-generated texts in form of reviews are implicit data for the recommender system because they are potentially rich resource of both feature/aspects of the item, and the user's evaluation/sentiment to the item. Features extracted from the user-generated reviews are improved meta-data of items. In our work we see sentiments extracted from the reviews as user's rating scores on the corresponding features.

The recommender system that we have built uses KNN algorithm as its base model to give recommendations to the user. We will use the helpfulness factor of the review available in the dataset along with the feature tf-idf vectors extracted in the above sentiment analysis model as an input feature set to the KNN model. To improve the accuracy of the model we have considered only those reviews of the products for which at least 80% of the users have voted as "helpful". This was done in order to eliminate the fake / useless reviews from the dataset.

4. RESULTS:

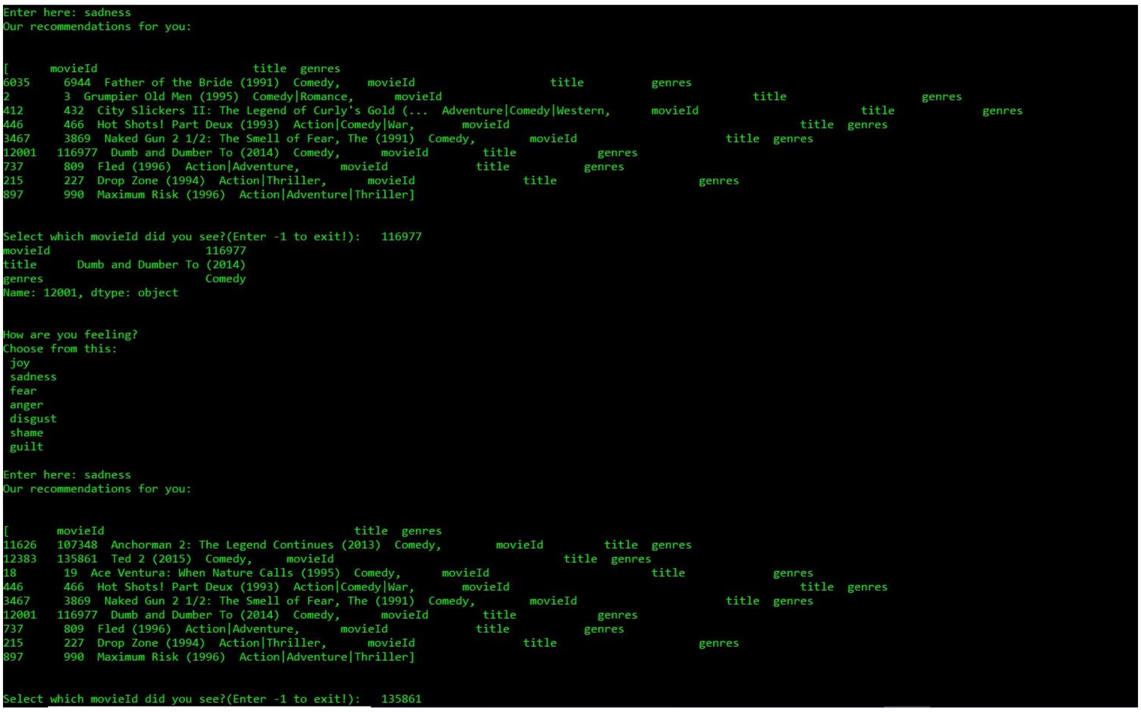
For the code that we submitted we have not included the 3rd party code [2] for the emotion detection. Hence, we are taking input from the user about their emotion as shown in the pictures above. In practice, this sentiment must come from analysis of user's text by sentiment analysis model.

Initially we gave “joy” and it suggested the movies that were initialized in the “happylist” bucket. We selected a movie which changed that bucket. Then for the next time, we mentioned “sadness” and so it again suggested the movies that were initialized in the “sadlist” bucket. We selected a movie by giving the movie ID and its bucket got updated accordingly. Then again we gave “sadness” as the emotion and the bucket then showed us new recommendations based on the movie that we last watched for the “sadness” emotion. We again watch a movie which would update its corresponding bucket and keep refining the recommendations according to our selections as explained before.



```
How are you feeling?
Choose from this:
 joy
 sadness
 fear
 anger
 disgust
 shame
 guilt

Enter here: joy
Our recommendations for you:


[      movieId                 title                  genres
7104     8950  The Machinist (2004)  Drama|Mystery|Thriller,      movieId                            title                 genres
564      593  Silence of the Lambs, The (1991)  Crime|Horror|Thriller,       movieId           title            genres
3795     4226  Memento (2000)  Mystery|Thriller,       movieId                  title        genres
7888    32291  Melinda and Melinda (2004)  Comedy|Drama,      movieId                  title                genres
269      281  Nobody's Fool (1994)  Comedy|Drama|Romance,      movieId                           title genres
517      538  Six Degrees of Separation (1993)  Drama,       movieId                 title          genres
5748     6559  Little Giants (1994)  Children|Comedy,      movieId                    title        genres
546      569  Little Big League (1994)  Comedy|Drama,       movieId                     title          genres
4598     5102  Rookie of the Year (1993)  Comedy|Fantasy]


Select which movieId did you see?(Enter -1 to exit!):   8950
movieId                      8950
title        The Machinist (2004)
genres     Drama|Mystery|Thriller
Name: 7104, dtype: object
```

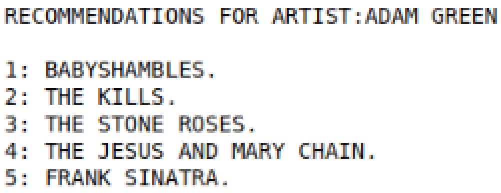
Figure 6: Movie recommendations for the emotion ‘joy’



```
Enter here: sadness
Our recommendations for you:


[      movieId                         title  genres
6035     6944  Father of the Bride (1991)  Comedy,    movieId                   title         genres
2           3  Grumpier Old Men (1995)  Comedy|Romance,      movieId                           title                       genres
412       432  City Slickers II: The Legend of Curly's Gold (...  Adventure|Comedy|Western,      movieId                 title                 genres
446       466  Hot Shots! Part Deux (1993)  Action|Comedy|War,       movieId                               title  genres
3467     3869  Naked Gun 2 1/2: The Smell of Fear, The (1991)  Comedy,        movieId                     title  genres
12001   116977  Dumb and Dumber To (2014)  Comedy,      movieId        title            genres
737       809  Fled (1996)  Action|Adventure,      movieId           title            genres
215       227  Drop Zone (1994)  Action|Thriller,      movieId                title                     genres
897       990  Maximum Risk (1996)  Action|Adventure|Thriller]


Select which movieId did you see?(Enter -1 to exit!):   116977
movieId                       116977
title      Dumb and Dumber To (2014)
genres                        Comedy
Name: 12001, dtype: object


How are you feeling?
Choose from this:
 joy
 sadness
 fear
 anger
 disgust
 shame
 guilt

Enter here: sadness
Our recommendations for you:


[       movieId                                    title  genres
11626   107348  Anchorman 2: The Legend Continues (2013)  Comedy,        movieId         title  genres
12383   135861  Ted 2 (2015)  Comedy,     movieId                           title  genres
18          19  Ace Ventura: When Nature Calls (1995)  Comedy,      movieId                   title            genres
446        466  Hot Shots! Part Deux (1993)  Action|Comedy|War,       movieId                               title  genres
3467      3869  Naked Gun 2 1/2: The Smell of Fear, The (1991)  Comedy,        movieId                     title  genres
12001   116977  Dumb and Dumber To (2014)  Comedy,      movieId        title            genres
737        809  Fled (1996)  Action|Adventure,      movieId           title            genres
215        227  Drop Zone (1994)  Action|Thriller,      movieId                title                     genres
897        990  Maximum Risk (1996)  Action|Adventure|Thriller]


Select which movieId did you see?(Enter -1 to exit!):   135861
```

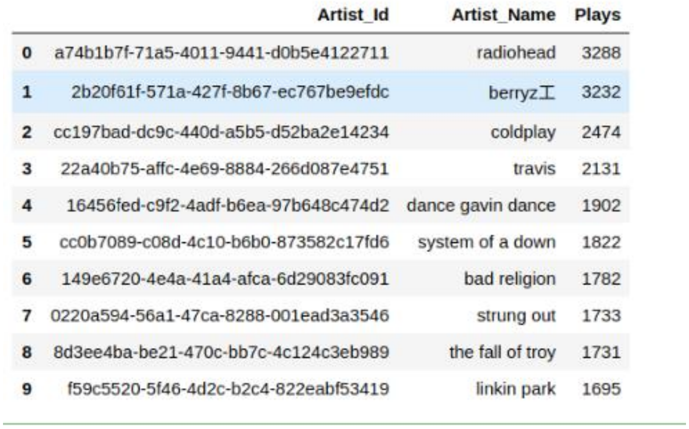
Figure 7: Recommendations for ‘sadness’

As we used both user-based and item-based approach we got two results for our music recommendation system. In the first result we can see item based approach see the model recommends artists similar to Adam Green and in the second approach we can see recommendation for a specifies user based on his gender, age and country.



```
RECOMMENDATIONS FOR ARTIST:ADAM GREEN

1: BABYSHAMBLES.
2: THE KILLS.
3: THE STONE ROSES.
4: THE JESUS AND MARY CHAIN.
5: FRANK SINATRA.
```

Figure 8: The item-based approach recommends music recommends artists like Adam Green.



	Artist_Id	Artist_Name	Plays
0	a74b1b7f-71a5-4011-9441-d0b5e4122711	radiohead	3288
1	2b20f61f-571a-427f-8b67-ec767be9efdc	berryz工	3232
2	cc197bad-dc9c-440d-a5b5-d52ba2e14234	coldplay	2474
3	22a40b75-affc-4e69-8884-266d087e4751	travis	2131
4	16456fed-c9f2-4adf-b6ea-97b648c474d2	dance gavin dance	1902
5	cc0b7089-c08d-4c10-b6b0-873582c17fd6	system of a down	1822
6	149e6720-4e4a-41a4-afca-6d29083fc091	bad religion	1782
7	0220a594-56a1-47ca-8288-001ead3a3546	strung out	1733
8	8d3ee4ba-be21-470c-bb7c-4c124c3eb989	the fall of troy	1731
9	f59c5520-5f46-4d2c-b2c4-822eabf53419	linkin park	1695

Figure 9: The user-based approach recommends music recommends artists based on user profile.

For the merchandise recommendation model, we ran our model on different settings. Firstly, we tried to vary the training and the testing set size (85% - 90% training data) and also the k value for KNN from (3 - 5). And the best results were obtained when the setting was training set of 85% and the k value = 5.

	Precision	Recall	F1- score	support
3	0.50	0.33	0.40	9
4	0.93	0.97	0.95	31
Total/avg	0.89	0.91	0.90	40

Table 4: Accuracy scores for merchandise model

The accuracy observed was 0.906 and the MSE was 0.094.

CONCLUSION AND FUTURE WORK:

We created a recommendation system that would incorporate sentiments derived from a text which could be user's post on a social media or user's comment/reviews. So far, we have implemented three models: for music, movies and merchandise. As we saw the movies model can read text, derive sentiments from the text such as happy, sad etc. and then can suggest movies from the genre appropriate for that emotion. The model picks up a movie from the pool of movies for the emotion suggested by text and then suggests 9 similar movies. If the user picks up a movie that was not suggested by the model, then we add that movie in that pool for that user making the model more user friendly. For the music recommendation model also, we have implemented item and used based collaborative filtering. Item based suggests 5 similar artists for an artist and user-based filtering suggests recommendation after looking at the user's profile, mainly its age, country and previous music history. The merchandise recommendation model reads product reviews and then gives a score to the product which is used in recommending the product to a new user.

To extend this model further we plan to make a similar system for recommending books which will use item-based filtering along with sentimental analysis from reviews. Further, we also aim to merge all these models and make one recommendation system where if we derive an emotion from any text the model could suggest us movie, music, books or merchandise. Also, if the user doesn't like what the system has suggested he would have the option to select any other item from the database. The newly selected item will then be added to the user's suggestion pool for that emotion. Like this user's preferences can replace system's recommendations so that every user has a personalized recommendation pool in our database.

REFERENCES:

[1] Sentimental Analysis: nearly everything you need to know. https://monkeylearn.com/sentiment-analysis/

[2] A python code to detect emotions from text. Author Rane, Poorva https://github.com/PoorvaRane/Emotion-Detector

[3] MovieLens. https://grouplens.org/datasets/movielens/

[4] Music recommendation Datasets for Research. Last.fm Dataset – 360K users. https://www.dtic.upf.edu/~ocelma/MusicRecommendationDataset/lastfm-360K.html

[5] Music recommendations with collaborative Filtering and Cosine Distance https://beckernick.github.io/music_recommender/

[6] Recommender System using Apache Spark. https://github.com/nachiketmparanjape/Music-Recommender-last.fm

[7] Amazon Data. https://www.kaggle.com/druss4/amazon-data

[8] Michael D. Ekstrand, John T. Riedl and Joseph A. Konstan (2011), "Collaborative Filtering Recommender Systems", Foundations and Trends® in Human–Computer Interaction: Vol. 4: No. 2, pp 81-173. http://dx.doi.org/10.1561/1100000009

[9] Pazzani M.J., Billsus D. (2007) Content-Based Recommendation Systems. In: Brusilovsky P., Kobsa A., Nejdl W. (eds) The Adaptive Web. Lecture Notes in Computer Science, vol 4321. Springer, Berlin, Heidelber

[10] Machine Learning-Based Sentiment Analysis for Twitter Accounts Ali Hasan 1, Sana Moin 1, Ahmad Karim 2 and Shahaboddin Shamshirband 3,4,*