Report on



MOVIE RATING PREDICTION USING REDDIT DATASET

JANMJAY SAHOO

NAGMA NISHAT

SAQUIB KHAN

SUNNY ANAND

SWASTIK SABAT

Contents

1. Design Document
   1. Summary of problem
   2. Description of Data
   3. Big Data Strategy
   4. Expected Results
2. Source Code
3. Analysis of Results(Validation)
4. Conclusion
5. Roles of team members

Design Document

**Summary of problem:**

The idea of Big Data seems so exciting when we hear it, huge amount of data ,using them to find those hidden information and then using them to create some real value add to our existing offerings. Big data processing seems to play such an important role in e-commerce and online offerings, which inspired us to make use of Big Data. Most movie reviews ask people to come up with a number (even though sites like IMDB allow you to write a review, in the end it asks you to select a number between 1 and 10). Sometimes, the words are not converted to numbers fairly. So we decided to rate movies based on people's comments about them.

Movie reviews are based on the people choice of a number in the range of 1 -10, however the true intention which lies in the comments is hardly captured. This accurate measure of a user review can be better captured by converting the words into an appropriate measure of rating for a movie. The problem of movie rating system to be based on a single measure makes it difficult to be as accurate as it can be.

**Assumptions:** The sarcastic comments will be assumed to be simple text and would not be a determining factor in sentiment analysis for movie rating using the user comments.

**Limitations:**  The Reddit Dataset doesn’t provide for movie rating on a scale of 1-10 like IMDB, it only provides for comments. So analyzing the comments to predict a rating for the movie using Reddit dataset and using the IMDB dataset for comparing that movie rating is a limitation.

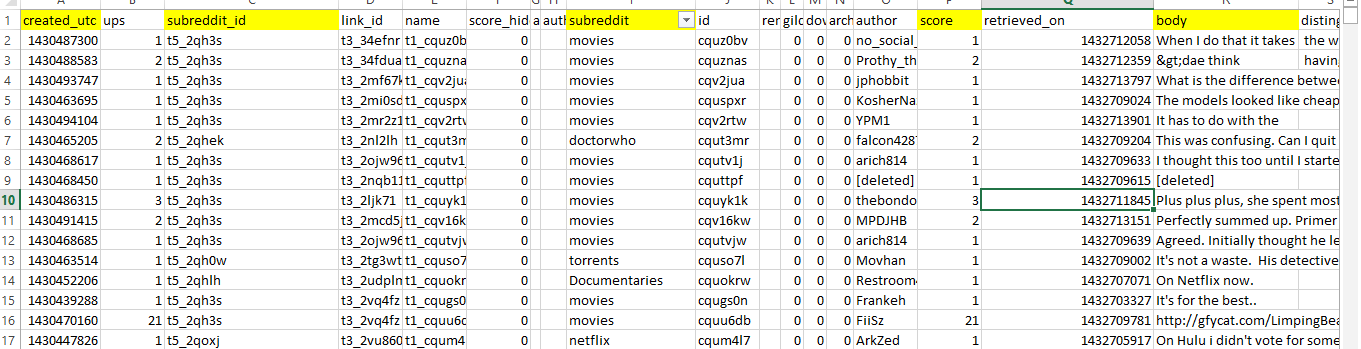
**Description of Input REDDIT DATASET:**

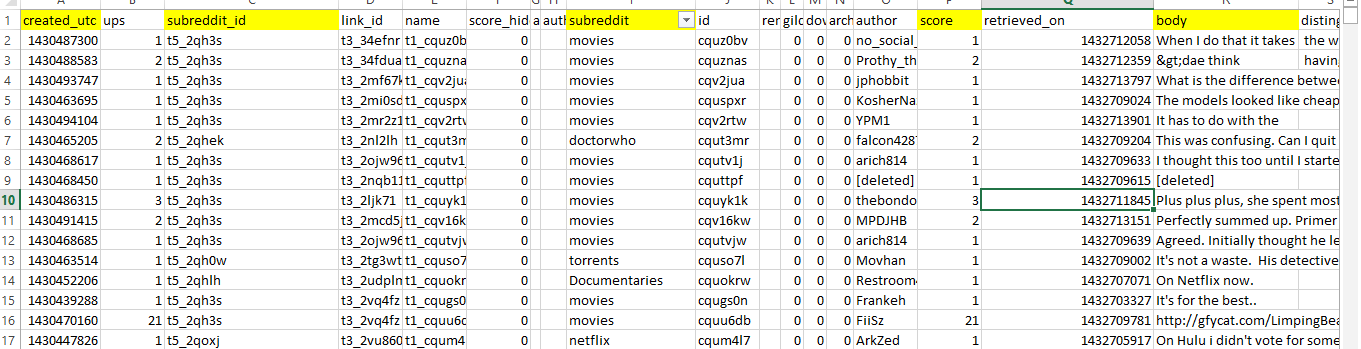
* Archive of Reddit comments from October of 2007 until May of 2015
* Each file is bzip2 compressed.
* Each uncompressed file is a series of JSON Blocks.
* Compressed size is 163G, Uncompressed file size is 553G on disk, 908 GB of JSON Documents.
* One of the most popular websites where people talk about everything is Reddit.com
* Reddiquette: One person starts a topic, others comment, people upvote or downvote the comments.

Example of REDDIT DATASET

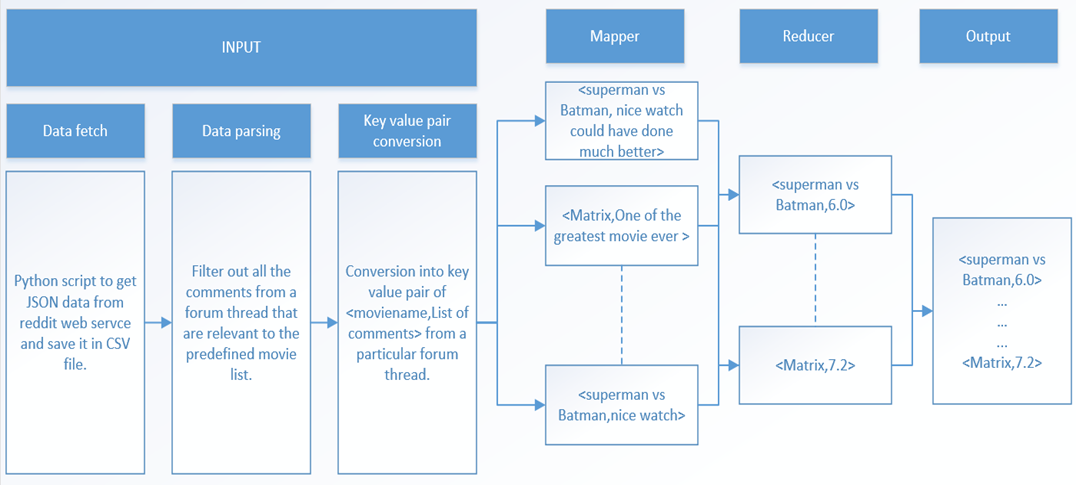
****

**Actual Dataset**

****

****

**Data Flow Diagram Using Example:**



**Pseudocode:**

**Step 1**

Data retrieval: Python script is used to get JSON data from Reddit web service and converted of Json data into CSV file and save it.

**Step 2**

Data Filtering: From the list of movies in the movieList files filter out the relevant comments that are present in the movieList file.

**Step 3**

Conversion: into key value pair: Data set is segregated to convert into key value pair with movie as key name and the comment as the value.

**Step 4**

Mapper: Key value pair is mapped into the mapper for further processing.

**Step 5**

Reducer: The score is computed for each score to give the final output as the score received by the movie.

**Explanation:**

1. **Preprocessing of data:**

a. We used a python script to get the Reddit dataset from a sqllite database that returns data in json format.

b. We stored this data in a comma separated file in the same python script.

c. You can find this code in the file Conversion.py which stores the entire data with all its attributes in a test.csv file.

2. **Filtering of data:**

a. We filtered our data based on a predefined list of movies.

b. For all the movies in our pre-defined list, we emmited a list of key value pairs of <commentID, list of comment on that forum>.

c. We filtered these key-value pairs to pairs of <movie-name, list of comments relevant to that movie>.

3. **Map Phase:**

a. We emmitted a key-value pair of <movie-name, score of each comment about the movie>.

b. The score was calculated using NLTK library's Naive Bayes Classifier and score attribute in the dataset was used to train the classifier.

c. We used a regular Naive Bayes Classifier and two extensions of Naive Bayes Classifier(K-Means Naive Bayes Classifier and Bigram Collocations Classifier).

4. **Reduce Phase:**

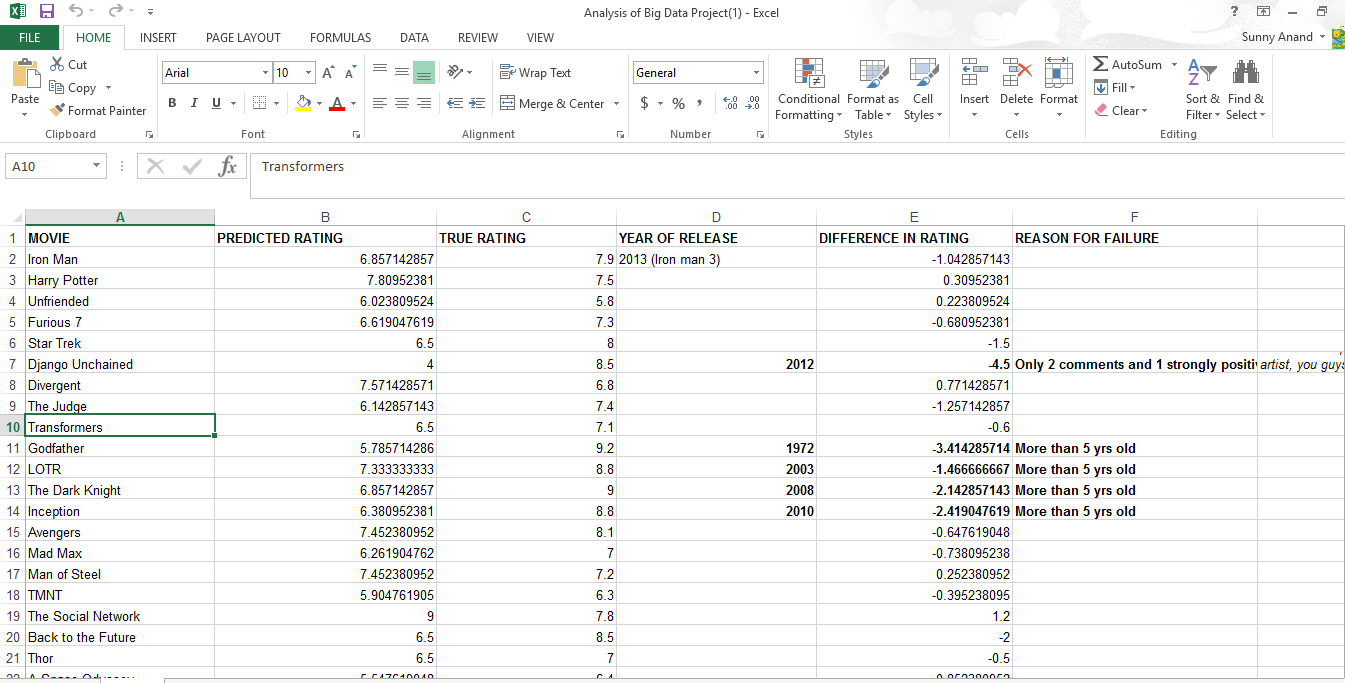
a. We filtered the scores based on individual number of comments about the movie being more than 5.

b. We reduced the key-values pairs to a list of <moviename, cumulative score of all comments about the movie>.

5. **Output:**

a. The output of the reduce phase is compared to the IMDB scores of the movies and the root mean square error is calcualted for every movie.

b. We do this calculation for the output of all three classifiers and compare the results.



**Big Data Strategy:**

We are planning to use map-reduce strategy.

**Why this strategy makes sense?**

The Reddit dataset classifies to be a big data problem due to the 3 V’s associated with its dataset.

The choice of technologies vary based on the planned inference study.

Lots of pre-processing, sentiment computation needs to be performed. Hence, using mapreduce was a good tool to use in this context.

We used PySPARK with NLTK for Classification problem of large unstructured data with a load of features to choose from., which is what we chose in our big data strategy. We used Naïve Baye’s , Naïve Baye’s K-Means, Naïve Baye’s Bigrams classification from NLTK for predicting the movie rating which we get from the JSON database during data collection.

**TOOLS USED:**

* SPARK-PySpark
* SPARK –MAP REDUCE
* NLTK Package
* Naive Bayes Classifier & K-NN Classifier
* R- Modeling and Accuracy Analysis for Classifiers

**Details of Expected Result:**

As shown above from reducer output, the format is (Predicted Rating, True Rating) for a given movie.

**Handling Bad Data:**

K- means clustering we randomized the seeds for clusters to avoid being trapped in the local minima.

For sparse data for movie score we used our own correction factor of total number of negatives/total number of examples.

Analysis of Results

The validation of the report from sentiment analysis is done based on the commets by reddit users on various categories of movies. The data is downloaded from a sqllite database in json format. This data is saved in a csv file using a python script. The final data is written to csv file as well. This final-output file contains two columns: movie-name, rating calculated based on comments by reddit users. By having a look at this ratings and comparing them to imdb we can easily identify a proximity between them. This analysis is done for all the 3 classifiers we implemented for our system.

1. Naïve Bayes Classifier
2. K-Means Naïve Bayes Classifier
3. Bigram Collocations Naïve Bayes Classifier

For our project, we had tested our output against imdb movie ratings. The root mean square was better as we progress through the list of classifiers we have implemented. We found that the ratings we obtained from reddit were similarly distributed. However, for popular movies rated higher than 7.5 on IMDB, we found that our ratings were in the range of 6 to 7.5. We postulate that this is due to the sarcastic comments in forums for that movie. However, this distribution was very accurate for less-popular movies.

Thus, sentiment analysis can be used to identify movie ratings which can be termed comparable to heavily moderated website like IMDB.

Another advantage of this method of sentiment analysis is that, we can see see a change in public reaction to the movie as it gains or loses popularity, which will be important for some consumer product companies. Marketing companies can use this analysis to decide if they can change the marketing strategy of movies and what their customer-demographic as regards to reaction to the movie.

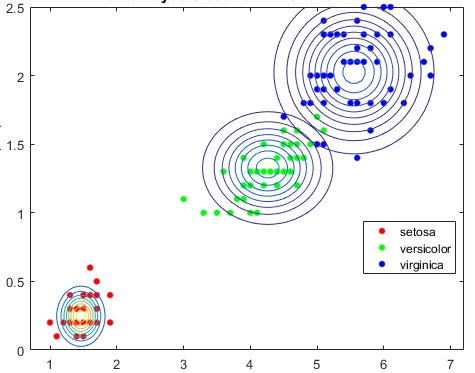
The major disadvantage of this method is that we are relying on Reddit forums to analyze the sentiment. Such a forum is filled with posts about popular movies and yet relatively unknown movies are mostly ignored. Another main disadvantage is that, social networking sites like Reddit mainly provide customer feedback, but not the movie critic reviews.

Naïve Bayes Classifier

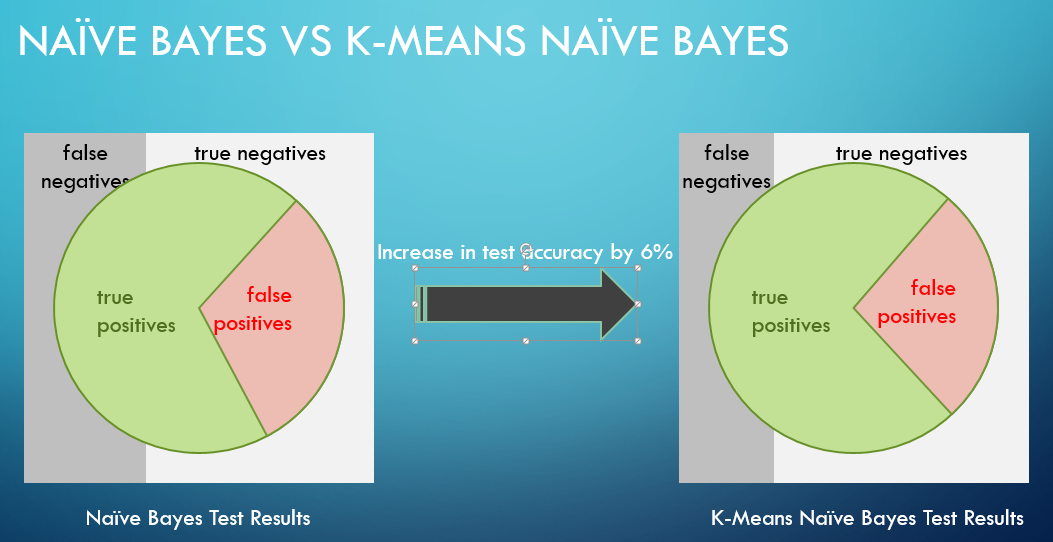
* Uses a bag of words model i.e. sequence of word occurrence irrelevant.
* We calculate two parameters for every word in our vocabulary:  
  P(comment|class=Pos) = ∏\_(𝑤∈𝑎𝑙𝑙 𝑤𝑜𝑟𝑑𝑠 𝑖𝑛 𝑡ℎ𝑒𝑐𝑜𝑚𝑚𝑒𝑛𝑡)▒"P(w|class=Pos)" \*P(class=Pos)
* P(comment|class=Pos) = ∏\_(𝑤∈𝑎𝑙𝑙 𝑤𝑜𝑟𝑑𝑠 𝑖𝑛 𝑡ℎ𝑒 𝑐𝑜𝑚𝑚𝑒𝑛𝑡)▒"P(w|class=Neg)" \*P(class=Neg)
* It assumed all features(words) as independent (called Naïve Bayes assumption).
* Single iteration classifier. Hence, considerably faster than others.
* Known to have respectable performance in text classification.
* The test and training examples were split 20:80 and the test set accuracy reported was almost 70%.
* The Univ. of Pittsburgh researchers found 82% agreement between two individuals when conducting tests for phrase-level sentiment polarity.
* Conclusion: A simple Naïve Bayes classifier ain’t bad!

K-Means Naïve bayes classifier:

* K-Means to create clusters that partition data into k clusters.
* Apply Naïve Bayes Distribution on each of these clusters:  
   K clusters ≈ K Naïve Bayes Distributions
* Randomized seeds, terminating condition of minimum squared error and repetition of 20 to avoid local optimum.

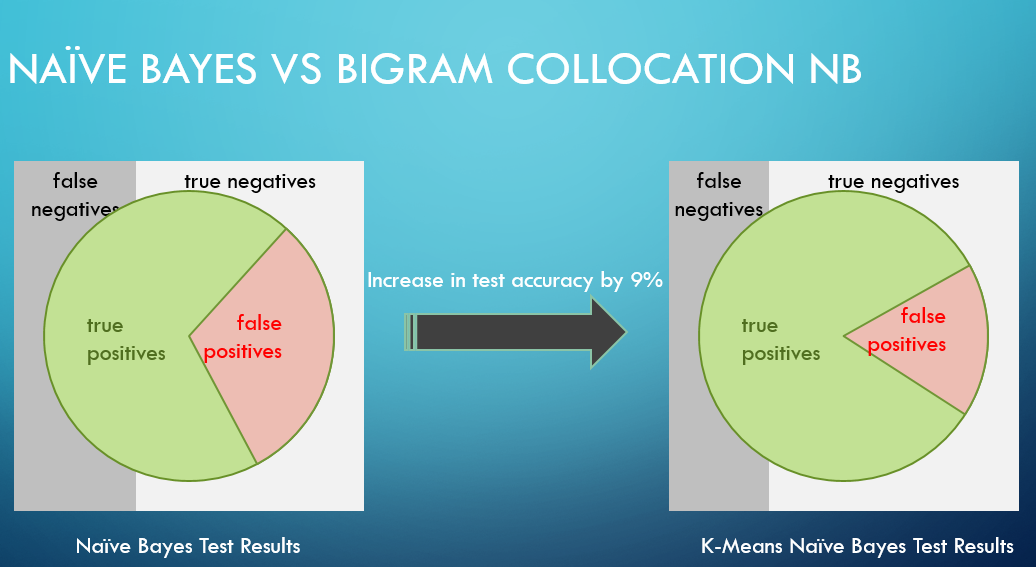


* Change in accuracy vs different values of k



BIGRAM COLLOCATIONS IN NAÏVE BAYES:

* What is a Bigram Collocation?
* Bigram Collocation measures collocation correlation of 2 words(frequency of the bigram vs frequency of individual words)
* A bag of words model sees ‘great’ as a separate word from ‘not’. Evaluates them separately. Hence, classifies the comment as positive.
* A bigram considers correlation of two adjacent words. Hence, evaluates “not great” together.



RMSE for Naïve Bayes Classifier =2.9,RMSE for K – Means Clustering =2.4,

RMSE for Bigram Collocation =2.23

BIGRAM COLLOCATION NAÏVE BAYES

Accuracy over test set: 78.4%

Conclusion

**How Big Data helped in this project?**

For a single company for a short time frame, we collected huge amount of data, for example apple had generated 40k tweets, and for this use case using mapreduce in terms of spark was an efficient technique with a in memory processing ability combined.

Lots of pre-processing, sentiment computation needs to be performed. Hence, using mapreduce was a good tool to use in this context.

**Describe briefly what you learnt in this project:**

* How to select predictor variable from a given data set
* Comparison between Two classifiers on a single dataset
* How to use PySPARK to handle Big Data methods.

**Describe how this project can be improved?**

* Review by user demographic can be used to improve the movie rating and hence predict a better movie rating.
* We can reduce the historical date range to something like a day to make better prediction using most recent comments.

Roles of each team member

DATA COLLECTION & ANALYSIS- JANMJAY &NAGMA

PySPARK PREPROCESSING & NLTK- SWASTIK & SUNNY

CLASSIFIER MODELING – SAQUIB & SUNNY

CLASSIFIER COMPARISON AND MOVIE RATING- JANMJAY & SWASTIK

NORMALIZATION & CLASSIFIER COMPARISON- NAGMA & SAQUIB