SMAI Course Project

Predicting Movie Ratings Based on Reviews

***Team**: 3*	***Project**: 9*	***Mentor**: Satyam Mittal*	***Faculty**: Dr Ravi Kiran*

ANANYA MUKHERJEE (2018801009)
HEMA ALA (2018701021)
RITESH (2018900060)
SHIVANI SRI VARSHINI (2018801014)

Literature Survey:

http://aclweb.org/anthology/Y07-1050 : In this paper they have conducted various experiments on different classifiers like SVM, Maximum Entropy and Scoring. Then they have collaborated all the three models which significantly increased the performance.

Two Methods of Integration:

1. Naïve Voting
2. Weighted Voting

Data Extraction:

- Download the IMDB dataset from Kaggle (https://www.kaggle.com/orgesleka/imdbmovies/version/1)
- Perform Data Cleaning (missing attributes/ misplaced data)
- From the CSV file we can extract the movie title id for each given movie.
- Frame a url https://www.imdb.com/title/**tt0032976**/reviews
  For example, title id of movie Rebecca is **tt0032976**.
- Using web scrapping, we can retrieve the User Reviews section from the above mentioned url.
- Storing the reviews and rating as a dict with title id as key using 'pickle'

Problem Categorization:

It is a classification problem. The comments are classified into 10 classes (1-10). Later average is calculated for the predicted rating which results in the final movie rating.

Success Metric: Accuracy,Precision,Recall,F1-Score

Feature Extraction: Review to TFIDF vector

Paper Implementation

Dataset:

fn	tid	title	wordsInTitle	url
titles01/tt0012349	tt0012349	Der Vagabund und das Kind (1921)	der vagabund und das kind	http://www.imdb.com/title/tt0012349/
titles01/tt0015864	tt0015864	Goldrausch (1925)	goldrausch	http://www.imdb.com/title/tt0015864/

Sample review after scrapping:

['Its a great movie undoubtedly and a must watch atlas watch it before you die you may learn something you never wanted to miss Its / for one of the finest silent movies ever', '10']

Binary Label as per paper:

Classified the dataset into positive and negative labels based upon the rating keeping the threshold as 7. A review having rating above 7 is positive, else negative.

Multiple Classifier:

Implemented the below classifiers to classify the movie reviews

1. **SVM**
   SVMs are a machine learning algorithm that was introduced by Vapnik (1999). They have been applied to tasks such as face recognition and text classification. An SVM is a binary classifier that finds a maximal margin separating hyperplane between two classes. The hyperplane can be written as: $y = w.x + b$
   where x is an arbitrary data point, i.e., feature vectors, w and are decided by optimization. The instances that lie closest to the hyperplane are called support vectors.
   We use SVM $y \in \{ -1,1 \}$ package for training and testing, with all parameters set to their default values.
2. **Maximum Entropy**
   Maximum entropy modelling (ME) is one of the best techniques for natural language processing (Berger et al., 1996). We used library to train Maximum Entropy model.
3. **Scoring**
   Scoring is based on word polarity scores calculation. If the final score is greater than zero, then the comment is positive else negative.

Integrated Classifier:

Integration of the above implemented classifiers are done using two methods.

1. **Naïve Voting**
   As the final output, we use the majority vote from three classifiers, namely SVMs, ME and Scoring methods.

2. **Weighted Voting**

This method uses each distance from hyperplanes of each classifier as weights (confidence) of the outputs. However, ranges of each distance computed from each classifier are not equivalent. We normalize each distance as follows.

Scoring: The actual value of the output from the classifier.

SVM: dist(d) x l

ME: (p(positive,d) - p(negative,d)) x m

dist(d) where is the distance from the hyperplane and are the probabilities of a document d as positive and negative opinions. l and m are constant numbers for the normalization. The values are computed beforehand from training data. In other words, the values are determined from the results obtained from training data by using each classifier constructed from it. l and m in this paper are based on the average of the distance from the hyperplane in the classification results.

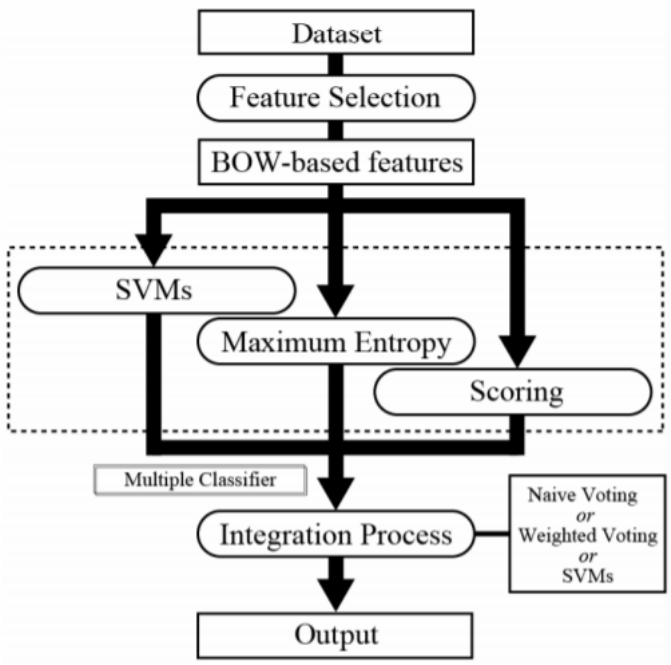


Figure. 1.The outline of our method.

Performance:

SVM:

	precision	recall	f1-score	support
0	0.91	0.84	0.88	382
1	0.85	0.92	0.89	380
avg / total	0.88	0.88	0.88	762

ME:

	precision	recall	f1-score	support
0	0.95	0.84	0.89	382
1	0.86	0.95	0.90	380
avg / total	0.90	0.90	0.90	762

Scoring:

	precision	recall	f1-score	support
0	0.86	0.44	0.58	382
1	0.62	0.93	0.75	380
avg / total	0.74	0.68	0.66	762

Weighted Voting

	precision	recall	f1-score	support
0	0.97	0.79	0.87	382
1	0.82	0.98	0.89	380
avg / total	0.90	0.88	0.88	762

Our Approach:

- Extending it as a multiclass problem (1-10)
- Predict individual review comment rating.
- Calculate movie rating by taking average.

Dataset:

- Consider the data-set consisting of reviews and ratings together which was initially obtained.
- Ignored the comments which do not have any rating in the IMDB website.

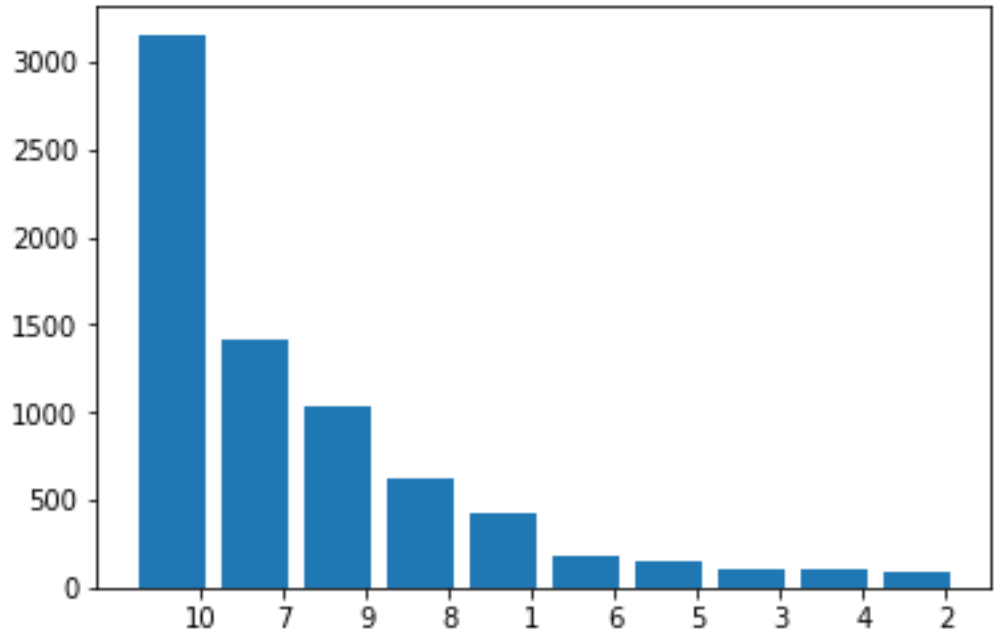
Models:

Used the below models using the existing library functions. We have trained the model on 295 movies each consisting of reviews along with rating.

- **LSTM**
- **SVM**
- **Naive Bayes**
- **KNN**

Class Imbalance:

Problem faced while implemented using LSTM. Due to class imbalance the learning was biased and resulted in 30% accuracy.



Results:

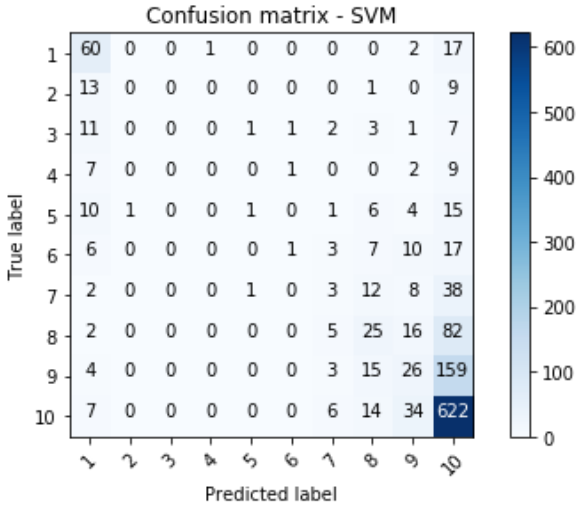
TITLE	Actual	SVM	KNN	NB
Salinui chueok (2003)	8.86	9.5	7.04	10
Vergiss mein nicht (2004)	7.65	9.1	8.73	10
Rang De Basanti (2006)	7.88	9.6	8.68	10
Das Schweigen der Lämmer (1991)	9.77	9.4	8.5	10
Oldboy (2003)	6.77	8.1	9.29	10

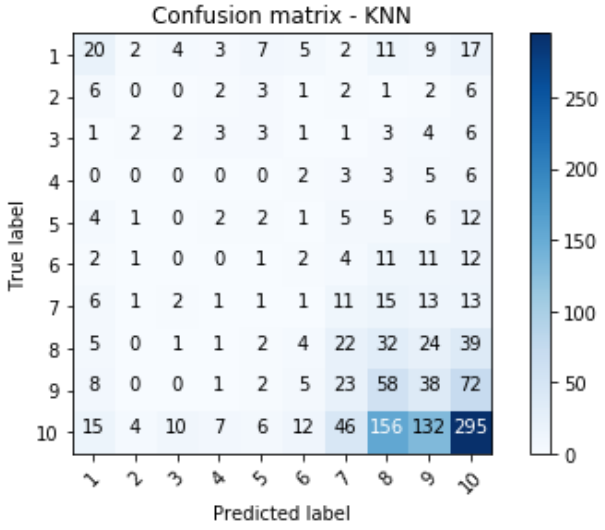
Performance:

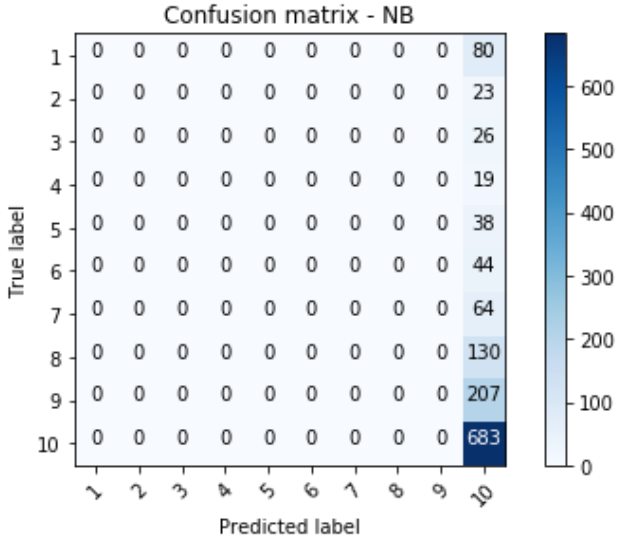
Training and Testing accuracy:

	Train	Test
SVM	87%	56.00%
Naïve Bayes	51%	51%
KNN	76%	30%

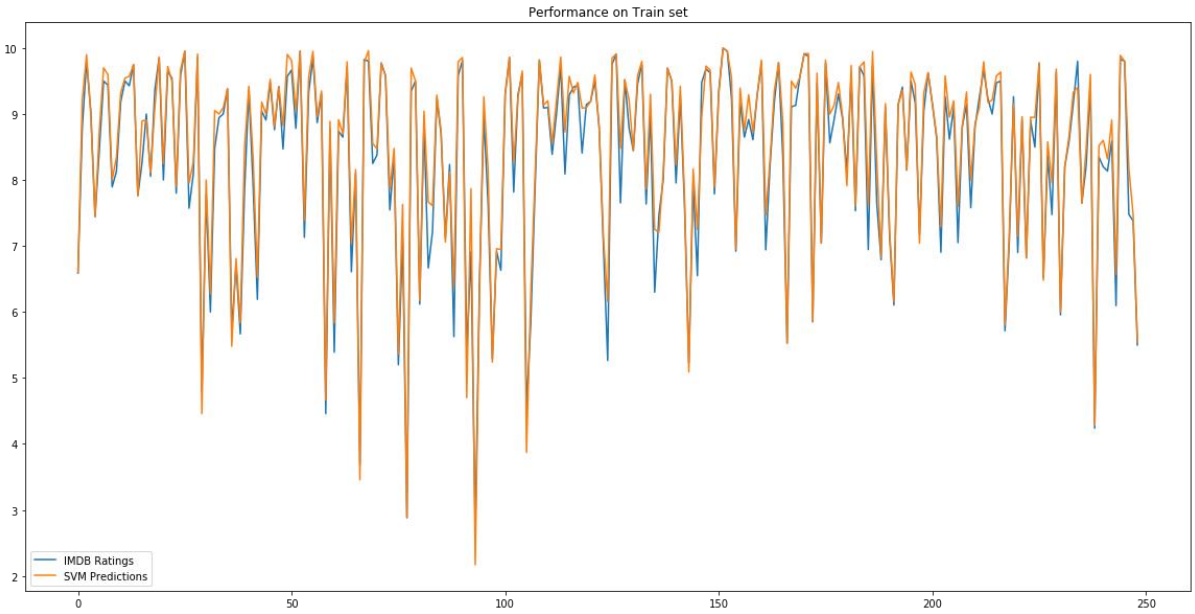
Confusion Matrices:

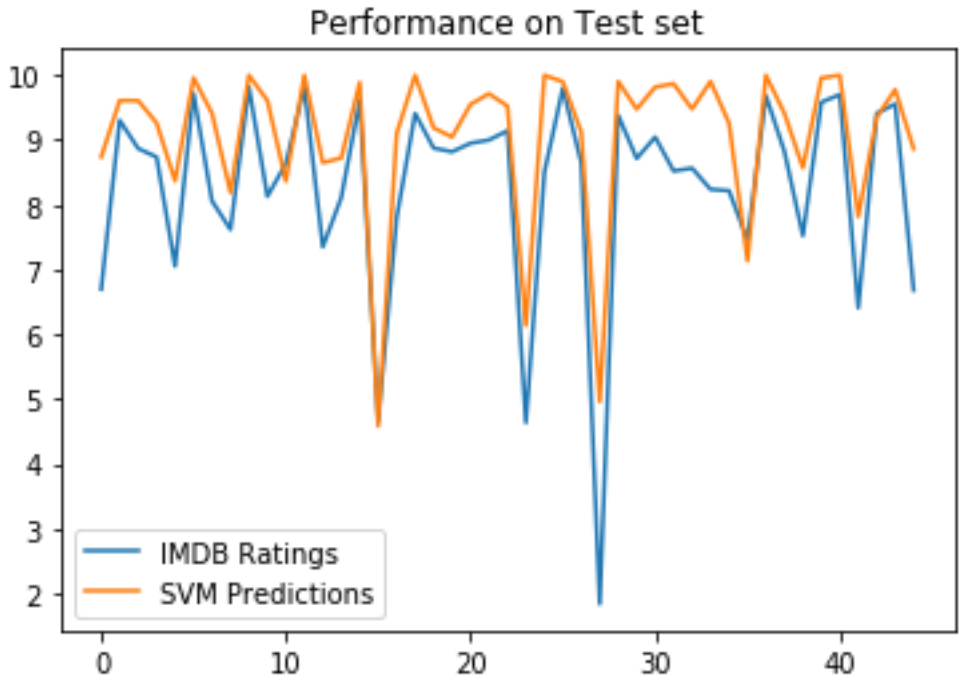






SVM Performance Visualizations:





Test Data Performance:

	Accuracy	Precision	Recall	F1-Score
SVM	57%	47%	57%	48%
Naïve Bayes	50%	25%	50%	34%
KNN	33%	40%	33%	35%

Movie Prediction based on different Categories:

```
Movie Title:  Rashomon (1950)
=====>Actual Rating: 8.473684210526315
=====>SVM rating: 9.052631578947368
=======>Acting: 8.421052631578947
=======>Direction: 9.444444444444445
=======>Song: 9.75
```

Book My Show:

In the IMDB dataset we have considered all the movie across all the country box offices which included Indian movies too. In order to show case, the Indian movies exclusively, we need Indian movies dataset. But web-scrapping bookmyshow.com is not legal.

If we see the robot.txt file of bookmyshow, it shows the following.

```
User-agent:*
Disallow: /
```

Hence our work is purely on the IMDB dataset.

Conclusion:

We have implemented three models as per the mentioned paper for binary classification of the movie reviews. Further integrated them using naïve voting and weighted voting methods.

We have come up with the implementation of our approach which initially included the training of LSTM model. We have observed that due to class imbalance data, the model was biased.

Hence, we adopted the machine learning techniques like SVM, KNN & Naïve Bayes. We observed that SVM performed better, it handled the imbalance data by using a linear kernel also by giving same importance (weightage to all the classes).

Our model also gives the prediction based on categories like Music, Direction & Acting.