Potential Movie Rating Prediction using Supervised Machine Learning Methods

Taslim Mahbub   
Computer Science & Engineering Dept. *American University of Sharjah*Sharjah, United Arab Emirates  
b00075270@aus.edu

.

Muhammed Yusuf Dada  
Computer Science & Engineering Dept.  
American University of SharjahSharjah, United Arab Emirates  
b00068047@aus.edu

.

Mohamed Shezan Rizny  
Computer Science & Engineering Dept.  
American University of SharjahSharjah, United Arab Emirates  
b00068372@aus.edu

*Abstract*—This electronic document is a “live” template and already defines the components of your paper [title, text, heads, etc.] in its style sheet. *\*CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract*. (*Abstract*)

Keywords—component, formatting, style, styling, insert (key words)

# Introduction (*Heading 1*)

This template, modified in MS Word 2007 and saved as a “Word 97-2003 Document” for the PC, provides authors with most of the formatting specifications needed for preparing electronic versions of their papers. All standard paper components have been specified for three reasons: (1) ease of use when formatting individual papers, (2) automatic compliance to electronic requirements that facilitate the concurrent or later production of electronic products, and (3) conformity of style throughout a conference proceedings. Margins, column widths, line spacing, and type styles are built-in; examples of the type styles are provided throughout this document and are identified in italic type, within parentheses, following the example. Some components, such as multi-leveled equations, graphics, and tables are not prescribed, although the various table text styles are provided. The formatter will need to create these components, incorporating the applicable criteria that follow.

# Literature Review

## Selecting a Template (Heading 2)

First, confirm that you have the correct template for your paper size. This template has been tailored for output on the A4 paper size. If you are using US letter-sized paper, please close this file and download the Microsoft Word, Letter file.

## Maintaining the Integrity of the Specifications

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

# Methods and Datasets

In this section, we shall look at the raw data available to us and select the attributes that are suitable to us. We will also examine how we selected the labels for the dataset. The dataset used for our project was taken from [Kaggle](https://www.kaggle.com/rounakbanik/the-movies-dataset) – the movies dataset.

The data was obtained by using the TMDB API to retrieve attributes of over 45000 movies. The main csv folder included the following attributes on 45000 movies that were used for our study:

* *Budget – in US dollars*
* *Genres – multiple genres per movie if applicable*
* *ID- identification number as index*
* *Popularity – quantified popularity measure*
* *Runtime – length of movie in minutes*
* *Vote\_average -the average of all the votes from users per movie – from 0 to 10 with 10 being the highest.*

The other attributes that could be potentially used were Revenue, number of Spoken Languages, and whether or not the movie belongs to a collection. To keep the experiment simple, we decided to only include the attributes listed.

## Data Preprocessing

At this stage, we shall describe the content that we extracted from our data to feed into the machine learning models. To preprocess and clean the dataset, we used the Pandas library in python. Firstly, we dropped all the columns that were not meaningful to us. Then, we converted the numerical columns to numerical data type, this included budget, runtime and popularity.

We then removed all then null values, from both numerical and non-numerical data columns. However, this was not sufficient since a lot of movies had a 0 value for the numerical data. We spotted this by plotting the distribution of all the quantitative columns. All the 0 values were removed from the dataset. Next, all the duplicate records were removed.

The genres column had multiple values per movie. In our experiment, we wanted to focus on the primary genre of each movie. Hence, the list of json objects were processed to only keep the first (primary) genre per movie.

Another important aspect of preprocessing that remains is categorizing the ratings into discrete labels. Currently, we had a rating between 0 to 10 for every movie. By logically considering the movie ratings, we decided to simplify the labels to Good (7.5+), Average (5 to 7.5) and Bad (0 to 5). This resulted in an imbalanced dataset, as expected, since majority of the movies would fall within the Average class. There were only 479 Good movies and 884 Bad movies, as compared to 5204 Average movies.

In an attempt to solve the imbalanced class problem, we attempted the following:

* Oversample the minority class labels (Bad, Good) to 2000 data points each.
* Under sample the majority class label (Average) to 2000 data points.

Over sampling the data can often lead to overfitting, while under sampling can result in valuable data features being lost in the process. To prevent a significant impact of either of these changes, we decided to compromise between the over and under sampling.

The remaining categorical column of genres were then one-hot encoded resulting in the following columns in our data frame: 'Action’, ‘Adventure', 'Comedy', 'Crime', 'Drama', 'Horror', 'Thriller'. The labels were saved for later analysis.

Finally, the numerical columns of 'budget', 'popularity', 'runtime', were normalized using the min-max normalization technique. It is imperative to normalize our data since some of the models are sensitive to the data scaling issue. With this, the data processing was complete and the result was saved into arrays.

## Feature Selection and Dimensionality Reduction

In addition to saving the raw preprocessed data, as mentioned above, we also applied feature selection and dimensionality reduction.

* First, the selectKbest method in SKLearn was used- which selects the k highest scoring features, where k was used as 5. The scoring function used was the chi-squared test, which measures the dependence between the random variables so using this function means that it would weed out the features that are likely to be independent of the class.
* Second, the selectPercentile method of SKLearn was used – it selects features according to a percentile of the highest scores. In our project, we selected to keep 50 percent of the data. For the scoring function, we used f\_classif, which calculates the ANOVA F-value between features for classification tasks. It measures if “the variance between the means of two populations are significantly different?”. Variances measure the dispersal of the data points around the mean.
* Third, we use PCA to select an appropriate number of components. As seen on Fig. 3.1 below, the 6th component only explains 6.5% which is very small. The first 5 components explain a total of 92.3%. Hence, we decided to only take the first 5 components, since the remaining do not explain a lot of variance.

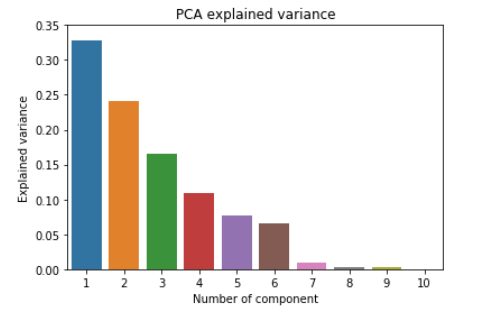


Fig. 3.1 Number of PCA components and their explained variance.

Overall, we ended up having 4 different types of input data for each of the models, the raw data with all features, selectKbest data with 5 featues, the selectPercentile data with 50% of useful features, and the PCA data with 5 principal components.

## Machine Learning Models

For each of the supervised machine learning models discussed below, we used a Grid Search to find the best combination of hyper-parameters for each of the 4 types of input data.

### Decision Tree classifier

The parameters tuned with their respective range:

* *Minimum samples required to split a node – range from 5 to 80 with increments of 5*
* *Maximum leaf nodes in the model – range from 5 to 80 with increments of 5*

### Naïve Bayes classifier

The parameters tuned with their respective range:

* *Var\_smoothing (portion of the largest variance added for calculation stability) – range from 1e-6 to 1e-15*

### KNN classifier

The parameters tuned with their respective range:

* *Weight function used in prediction – either ‘uniform’ or ‘distance’*
* *Number of neighbors to use for classification - range from 5 to 80 with increments of 5*

### SVM non-linear classifier

The parameters tuned with their respective range:

* *Kernel – either 'rbf' and 'sigmoid'*
* *C – a selection from the list: 0.1,1,10,100,1000*
* *Gamma – a selection from the list: 1e-4,1e-3,1e-2,0.1,1,5,10*
* *coef0 ( Independent term in kernel function) – range from -200 to 200, increments of 50.*

### SVM linear-kernel classifier

The parameters tuned with their respective range:

* *C – a selection from the list: 0.1,1,10,100,1000*

For each of the input data format to the models discussed, we used the train\_test\_split method from SKLearn with a fixed random state of 200 – to divide the data set into 80% training data and 20% testing. The random state ensures a consistent selection to increase fairness.

## Evaluation

For each of the models discussed, the evaluation was repeated for the 4 different input data types to the model (Default data, K best Data, K select Percentile Data, PCA data.). The evaluation steps included:

* The best model from the Grid Search was chosen based on the F1-Score (macro).
* The predicted labels were generated from the trained model and compared with the actual to get the testing score. The training score was also saved for the best model.
* A classification report indicating the precision, recall and f1-score for every class (Average, Good, Bad) were generated using the predicted and actual test labels.
* The confusion matrix was plotted.
* Using the best hyper-parameter selected model, we performed cross-validation with 5 splits. The resulting F1-Scores for every model was printed out to show the variation due to different selection of training/testing data.

# Experimental results

## Summary of Results

As mentioned earlier, we trained the 5 different supervised learning models (Naive Bayes, KNN, Decision Tree, Non-Linear SVM and Linear SVM) using 4 different types on input data (raw preprocessed, selectKBest, selectKPercentile and PCA) and randomly cross-validated all results (5-fold cross-validation). The standard for performance evaluation was the F1-Score. As expected, there was a variation in performance of a given model based on the type of input it received (4 different results for each model). The best result from each model is summarized in table 4.1 below

Table 4.1 Summary of Experimental Results

| # | Model | Subhead |
| --- | --- | --- |
| 1 | Decision Tree Classifier | 0.656956 |
| 2 | Naive Bayes Classifier | 0.547154 |
| 3 | KNN Classifier | 0.783097 |
| 4 | SVM Non-Linear Kernel | 0.651570 |
| 5 | SVM Linear Kernel | 0.579962 |

The KNN Classifier was found to be the best classifier with an F1 score of 0.783097. The KNN classifier produced this results when it considered the 20 nearest neighbor data points in its algorithm according to the distance similarity measure. The input data used to produce this result did not have any feature selection or dimensionality reduction applied.

A confusion matrix for this KNN classifier result is depicted below in Fig. 4.1. The confusion matrix shows the that the trained model classifies the test data into the correct rating class most of the time as indicated by the darker hues along the diagonal. The model works almost perfectly for the “Good” and “Bad” classes and fairly well for the “Average” class.

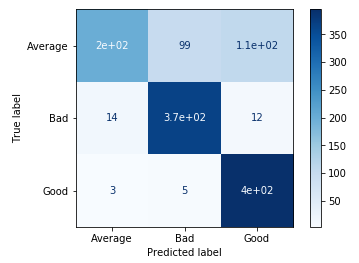


Fig. 4.1 Confusion matrix for overall best classifier – KNN

The cross validation also shows overall a very high score for each run using this particular classifier (range between 0.81 to 0.78), showing that this KNN model works well consistently.

The detailed class ification report is given below in Table 4.2

Table 4.2 Classification Report For Knn

| ***Class Label*** | Precision | Recall | F1-Score |
| --- | --- | --- | --- |
| Average | 0.92 | 0.49 | 0.64 |
| Bad | 0.78 | 0.93 | 0.85 |
| Good | 0.77 | 0.98 | 0.86 |

## Discussion of Results

Since we transformed our imbalanced dataset into a balanced one by applying a mix of over- and under- sampling, a lot of the data points from similar classes will be close together in a selective location. This helps explain why the KNN classifier performed so well in this experiment.

However, although we are able to detect the Good and Bad classes perfectly, we have to note that the Average class that was under-sampled has a considerably lower performance.

From table 4.2, we notice that for the “Average” data label we have the highest precision however the lowest recall. 92 percent of the points classified as “average”, actually were “average”. However, only 49% (less than half) of all real “average” movies were correctly classified by the model. For the remaining, classes of “Bad” and “Good” we have much better results as indicated by the higher recall and f1-score (harmonic mean of precision and recall).

The lower performance of “Average” class is likely due to overfitting which is indicated by the training score of this particular classifier being 1 - the KNN classifier can see a lot of the same points in a clustered region because they are replications of each other (Good and Bad classes were over-sampled). To overcome this issue, a future work could try to augment the data when over-sampling rather than simply replicating. This would add a bit of noise in the oversampled data, preventing the overfitting of the model.

# Conclusion

Machine learning is a powerful tool with a wide array of applications. However it is not all-powerful. The quality of the model trained and results produced is highly dependent on the quality of the dataset and how well the data is prepared.

This experiment and paper showcases how machine learning can be applied on a practical real-world dataset and use case. There are a number of steps involved in simply preparing the data like data exploration, preprocessing, feature selection, etc before even working with machine learning algorithms and techniques. This showcases the complexity of Machine Learning. Once that is done, the appropriate model or models need to be selected based on the form of the data and the research question. The hyper-parameters utilized by the complex algorithms need to be optimzed to produce the best results possible given the dataset. In this experiment, we implemented all the above steps at a fundamental level to produce results and better understand the data and interrelation between attributes.

The question we tried to answer with this experiment was, “What factor or attribute makes a movie successful”.

Based on the best results we obtained, as described in the previous section, we can extract the following information from the dataset:

1)

2)

3)

How machine learning solves problems in general

## Authors and Affiliations

**The template is designed for, but not limited to, six authors.** A minimum of one author is required for all conference articles. Author names should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

### For papers with more than six authors: Add author names horizontally, moving to a third row if needed for more than 8 authors.

### For papers with less than six authors: To change the default, adjust the template as follows.

#### Selection: Highlight all author and affiliation lines.

#### Change number of columns: Select the Columns icon from the MS Word Standard toolbar and then select the correct number of columns from the selection palette.

#### Deletion: Delete the author and affiliation lines for the extra authors.

## Identify the Headings

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is “Heading 5”. Use “figure caption” for your Figure captions, and “table head” for your table title. Run-in heads, such as “Abstract”, will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced. Styles named “Heading 1”, “Heading 2”, “Heading 3”, and “Heading 4” are prescribed.

## Figures and Tables

#### Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation “Fig. 1”, even at the beginning of a sentence.

1. Table Type Styles

| Table Head | Table Column Head | | |
| --- | --- | --- | --- |
| Table column subhead | Subhead | Subhead |
| copy | More table copya |  |  |

1. Sample of a Table footnote. (*Table footnote*)
2. Example of a figure caption. (*figure caption*)

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity “Magnetization”, or “Magnetization, M”, not just “M”. If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write “Magnetization (A/m)” or “Magnetization {A[m(1)]}”, not just “A/m”. Do not label axes with a ratio of quantities and units. For example, write “Temperature (K)”, not “Temperature/K”.

##### Acknowledgment *(Heading 5)*

We would like to extend our gratitude to Dr. Salam Dhou of the Department of Computer Science and Engineering at the American University of Sharjah, who instructed us on Machine Learning concepts and supervised this project. Her advise, feedback and support were vital to the realization of this work.

##### References

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

1. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
2. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
3. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
4. K. Elissa, “Title of paper if known,” unpublished.
5. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
6. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
7. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

**IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published.**