Movie Success Prediction

Project Report for

Machine Learning Course - COSC 6342

December 3, 2018

Students:

Mahsa Shafaei (1490000)

Hosein Neeli (1541673)

Guided by:

Prof. Dr. Ricardo Vilalta

University of Houston, Main Campus

Fall Semester, 2018

1. Why using ML is important in the movie industry?

Over the years, the number of released movies has increased massively (Krushikanth R Apala, 2013), but according to Internet Movie database (IMDB), only a few out of millions of movies get a high rating (higher than 8). Making movies is expensive and by predicting likability of movies, we can significantly affect the movie industry. For example, movies like “Jupiter Ascending”, “Valerian and the City of a Thousand Planets” and “The Lone Ranger” spent millions of dollars for production, but their IMDB ranking is less than 6.5 (which shows these movies are not very popular), and also, they could not make a profit in movie theaters. So, movie investors may lose a great amount of money by working on movies that are not liked by people. The cost of movie production comes from different sources such as production, marketing, screenings and financing costs. Our proposed method can be used as a tool by movie production companies to avoid most of these costs by early success prediction.

There are several works that introduce “Gross” value as a success criterion, and they tried to predict this value for the movies (gross value shows the amount of money that movie earned from the box office). In this work, we set our goal to automatically predict IMDB rating for the movies as a likability criterion because of four main reasons. First, the gross value is not available for a large number of movies. Second, the price of box-office ticket changes during the years, so we cannot compare old movies with newer ones. Third, this value is dependent on many other variables like movies advertisement and competitor movies. Finally, movie theaters are not the only source for movies revenue; there are other sources like home entertainment, television deals, and video on demand (i.e. Netflix and Amazon). So, by predicting likability of a movie, we can predict if selected movies by these companies will be popular among their users or not. Therefore, we use IMDB rating to define the success criterion, and we propose classification models to automatically predict this value for the movies.

Although intrinsic factors such as the story of the movie and the quality of scriptwriting play an important role in the likability of movies, extrinsic factors including the popularity of directors and advertisement are as important as the intrinsic ones.

In this project, we extract features from movie subtitles and combine these features with external features like movie genre, name of directors and actors. Then, we feed features to different machine learning models to predict the group of IMDB rating for movies. For classification models, we need to categorize movies to groups. We considered 6.5 as a threshold for this purpose (because with this threshold we can divide movies into two equal-size groups). Movies with rating higher than 6.5 are successful and movies with less than 6.5 are failure. In figure [1], you can see the diagram of system.

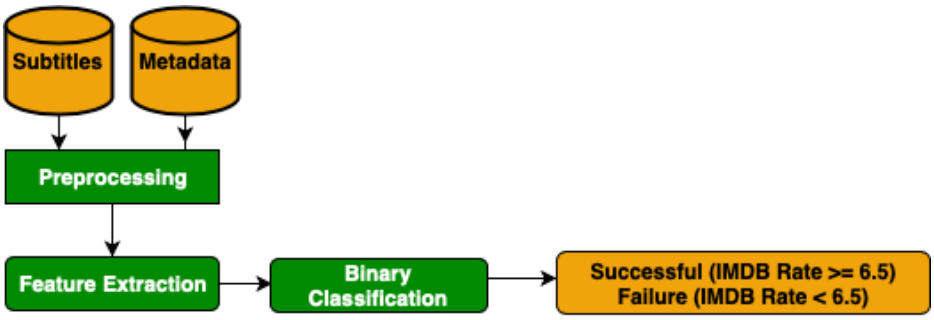


Figure 1 Diagram of the workflow of the system.

2. Feature selection

Besides external features like the cast and crew of the movie, we want to measure the effect of other features such as textual features in scripts.

2.1. Textual Features

Natural language processing (NLP) is a field in science that study how computers can understand human language (Chowdury, 2003). NLP inspires from various technics to understand and manipulate natural language, from artificial intelligent and machine learning algorithms, to technics similar to the way human being interact with each other. There are outnumbered applications of NLP such as speech recognition, text translation, text summarization, multilingual and cross language information retrieval (CLIR) and so on.

Referring to MIT survey (MIT-Tech-Review, 2017), 45 percent of major projects in renowned companies (in which machine learning is being used) are defined based on natural language processing or combination of NLP with other fields like image recognition/classification. 52 percent of responder are planning to define new project on NLP and its applications.

Here, the textual features we extract from movie subtitles are as follow:

2.1.1. N-Gram

LM or language models are models that assign probability to sequence of words. We could estimate joint probability of a sentence by calculating number of conditional probability of its words and multiply them together (based on chain-rule), but there is no way to find exact probability of a part of the sentence (word) given a long sequence of preceding words. The reason is that language is dynamic, and people use their sense of creativity to create sequence of words or sentences that have never happened before. So, simplify this problem, we use N-gram model (introduced by Andrei Markov, 1856-1922). For example, in bi-gram model, we calculate the occurrence probability of a word given previous words (conditional probability of

the preceding words). In N-grams we compute (N-1) past words to compute conditional probability. In this work we extract unigram, bigram features from text, and apply term frequency-inverse document frequency (TF-IDF) as the weighting scheme. We define cut-off threshold equal to 100, to avoid terms with lower document frequency less than this threshold

2.1.2. Skip-Gram

Skip Gram model is generalized model of the N-Gram model. The constraint of consecutive words in a sentence is removed in this model so we can jump over some words when we are calculating conditional probabilities. This model was introduced to address data sparsity problem of N-Gram model. We extract two skip-ngram 2 (n=2,3) features from text.

With the help of these features, we can extract authors' words selection.

2.2. External Features

2.2.1. Genre

Each movie is categorized in one or multiple class of categorization based on artistic, musical, story plot or literary composition characteristics. Genre helps movie fans to easily find what he or she likes to see. As each movie can have more than one genre, we used binary representation for showing the genre of each movie (bag-of-genres).

2.2.2. Actor and Directors

Two potential factors for movie success prediction are directors and actors. We employ these names as binary bag-of-names features.

3. Tools, Libraries and Environment

This project was done by means of Python programming language. Wide variety of tools and libraries were utilized for data preparation, training, testing and evaluation. Here you can see some of those important tools:

- **Scikit-learn**: One of the most powerful tools out there brings machine learning to python. It contains wide variety of sub libraries to train and evaluate models.
- **NLTK**: It stands for Natural Language Toolkit, contains series of libraries for Python to work with human languages.
- **PyPlot**: A useful library for data visualization in python.
- **Numpy**: Is one of the core Python libraries dealing with large collection of multi-dimensional arrays.

4. Models

We have trained 4 classification models for this project. In this section, we are going to represent comparison of SVM (Linear and Non-Linear), Decision Tree, K-Neighbor and Random Forest classifiers.

4.1.SVM

SVM stands for Support Vector Machine, is a supervised learning method which separates data with a hyperplane in N-dimensional space. N is the number of features we introduced to SVM. This algorithm tries to maximize the distance between hyperplane to nearest data points of each class. Nearest data points are called support vectors and the maximized distance is called margin.

4.2.Decision Tree

Decision tree classification algorithm is a relatively simple method to classify data in a hieratical approach (C, 2008). In decision trees, internal nodes represent a test or question on an attribute and each branch denotes answer of the question and each end node (leaf), contains final class of the test.

4.3.K Nearest Neighbor Classifier

This classifier uses majority voting of K most similar examples. KNN is considered as a supervised learning technique. Similarity is calculated based on distance metric (from observation to its neighbors). Based on characteristics of problems, other distance measurement like Hamming distance, Manhattan and Chebyshev can be used.

5. Performance Evaluation

In machine learning, training our own model and using it to predict result is pointless without evaluating the performance of the model. There are some conditions that we are in a situation that wrong classification costs us huge penalty. For example, in medical science, classifying a person as healthy person while he/she has cancer is unacceptable and has severe consequence. So, we are trying to evaluate our model and compare it with other models to have better understanding about how reliable it is. Here are some evaluation methods for machine learning classification models.

5.1.Splitting train and test data

We train our model with 80% of data and performance evaluation was done with rest 20% portion of data.

5.2.Confusion Matrix

Confusion matrix is the key resource for calculating performance of a classifier model. This matrix contains number of True Positive, True Negative, False Positive and False Negative predicted samples.

We use content of Confusion Matrix to calculate more complex evaluation parameters like precision, recall etc. Here you can see confusion matrix for all of the algorithms that we run on our dataset.

Table 1 Confusion matrix of Random Forest model.

Random Forest		Predicted	
		Failure	Success
Actual	Failure	TN = 47	FP = 48
	Success	FN = 40	TP = 177

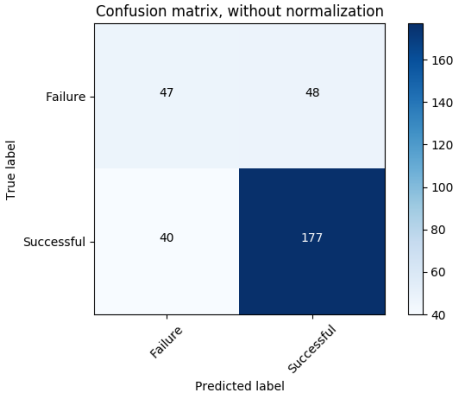


Table 2 Confusion matrix of linear SVM model.

SVM		Predicted	
		Failure	Success
Actual	Failure	TN = 44	FP = 51
	Success	FN = 27	TP = 190

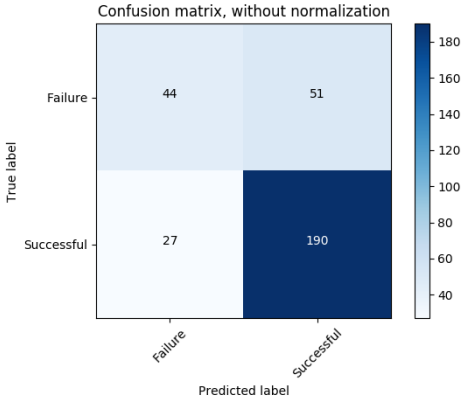
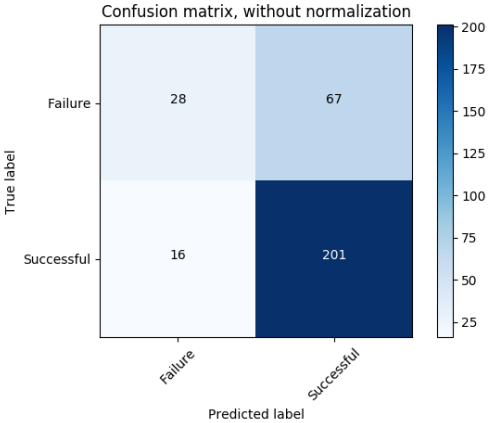


Table 3 Confusion matrix of KNN model.

KNN		Predicted	
		Failure	Success
Actual	Failure	TN = 28	FP = 67



	Success	FN = 16	TP = 201

Table 4 Confusion matrix of Decision Tree model.

Decision Tree		Predicted	
		Failure	Success
Actual	Failure	TN = 44	FP = 51
	Success	FN = 70	TP = 147

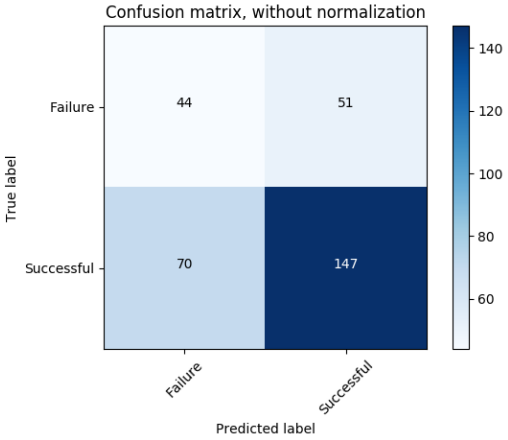
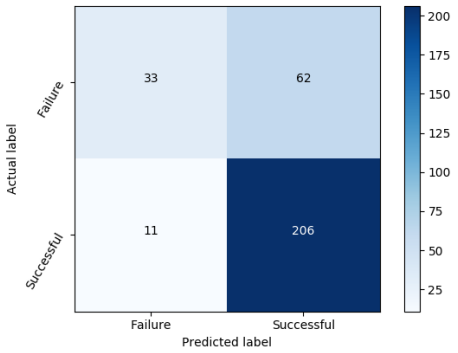


Table 5 Confusion matrix of non-linear SVM model.

SVM (non-linear)		Predicted	
		Failure	Success
Actual	Failure	TN = 33	FP = 62
	Success	FN = 11	TP = 206



5.3. Precision – Recall (Accuracy) Score

Accuracy

Accuracy is the simplest way to represent the performance of a model. It shows the ratio of observations that are correctly classified to the total number of observations. This parameter works well when the number of false negative and false positive classified objects are almost same. The following formula is used to calculate accuracy of a model: $\frac{TP+TN}{TP+TN+FP+FN}$

Table 6 Accuracy score comparison of different models.

Classifier	Accuracy Score
Random Forest	0.717

SVM (Linear)	0.750
SVM (non-linear)	**0.766**
KNN	0.733
Decision Tree	0.612

Precision

Precision is an evaluation parameter denotes how accurate our classifier model is out of predicted positives. This parameter is used when we would have high penalty for high False Positives. The formula which is used to calculate precision is: $\frac{TP}{TP+FP}$

Recall

This evaluation parameter represents how successful is our classifier to detect actual positives. The formula which is used to calculate precision is: $\frac{TP}{TP+FN}$. Recall is important where cost of False Negative is high.

Precision – Recall Curve

Precision Recall Curve (PRC) demonstrates precision scores for corresponding recall scores. Precision recall curve is one of the most powerful tools for evaluation of classification models. The area under the curve of PRC is used to compare performance of different classifiers. In the Figure [2] the Precision-Recall Curve of the provided models is shown.

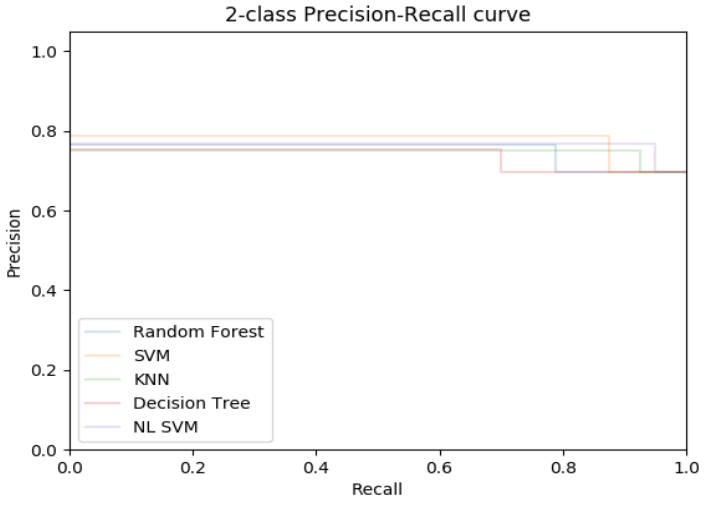


Figure 2 Precision-Recall curve of different classifiers.

The AUC (Area Under the Curve) of Precision-Recall Curve is used to compare performance of different classifiers by comparing one to one values of different thresholds for recall and precision. In table 7 you can see the AUC of different classifiers. The SVM model has higher area under the curve so it performs better that the other algorithms in our project.

Table 7 Area under the curve comparison of different models.

Classifier	AUC
Random Forest	0.77
SVM (Linear)	**0.78**
SVM (non-linear)	0.76
KNN	0.75
Decision Tree	0.73

5.4.F1 Score

F1 score is working based on putting a balance between precision and recall (Joshi, 2016). F1 score is calculated by weight averaging precision and recall. The formula is: $\mathbf{2} * \frac{\boldsymbol{Recall * Precisiion}}{\boldsymbol{Recall + Precisiion}}$. When there is no concern about cost of misclassification, F1 score is a good parameter to represent performance of our model.

There are three popular methods to calculate F1 score for multi class models. In the first method called Macro Averaging, we compute precision and recall for each class independently, then take the average to have final precision and recall scores. In this method if data is very imbalanced and our model is very bad in classifying smaller class, the final F1 score would still be affected by smaller class score. In Micro averaging we use sum of all values from confusion matrixes of all classes and then compute the final F1 score. In Weighted Averaging, we will consider occurrence of each class by assigning weight for computation of F1 score.

Table 8 F1 score comparison of different classifiers.

Classifier / F1 Score	Macro Averaging	Micro Averaging	Weighted Averaging
Random Forest	0.658	0.717	0.714
SVM (Linear Kernel)	**0.679**	0.750	**0.738**
SVM (non-linear)	0.662	**0.766**	0.735
KNN	0.615	0.733	0.699
Decision Tree	0.564	0.612	0.620

By looking at F1 scores in above table, we realize that **SVM** model performs better than other models. In cases that we face high dimensional space (high number of features), linear SVM model performs very well.

Generally, SVM classifier has good performance in presence of outliers because this classifier uses the most relevant to find support vectors. Other classifiers like KNN are more sensitive to outliers and one way to improve their performance is removing outliers. We have outliers in our data (e.g. bad movies with good cast and crew), but there is no way to exactly distinguish outliers in our case and we are not able to remove them, so it makes sense that SVM has better performance.

6. Conclusion

In this project, we used meta data and text related features to train our own classifiers to predict likability of a movie. From metadata we considered directors and actors of the movies and movie genres. From subtitle of the movies, we extracted unigram, bigram and k-skip grams. Mentioned features are pre-production features, thus based on our model's prediction, producer can estimate successfulness of their effort. Then, we used four different classification models (SVM (linear and nonlinear kernel), KNN , Decision Tree, and Random Forest). We evaluated our models with several performance evaluation parameters and found out SVM classifier performs better among all classifiers that we trained, with F1 weighted score of 0.73.

7. References

C, K. (2008). What are decision trees? *Nat Biotechnol*, (pp. 26(9): 1011–1013.).

Chowdury, G. G. (2003). Natural language processing. *Annual Review of Information Science and Technology*, (pp. 51-89).

Domingos, P. (2012). A Few Useful Things to Know about Machine Learning. *Communications of the ACM* (pp. Vol 55 Issue 10, Pages 78-87). Washington: University of Washington.

Joshi, R. (2016, Sep). Retrieved from EXSILIO Solutions: https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/

Krushikanth R Apala, M. J.-C. (2013). Prediction of movies box office performance using social media. *Advances in Social Networks Analysis and Mining* (pp. 1209-1214). ACM.

MIT-Tech-Review. (2017). *Machine Learning: The New Proving Ground for Competitive Advantage.* MIT Tech.