Powerful Model for Rating Prediction of Movies

1 Introduction

In the entertainment industry these days, movies are certainly one of the most powerful drivers of culture and economy as they are influencing billions of audiences in many aspects. In this digital era, with the development of streaming services and new media, the movie industry is undergoing innovation and all the production companies are facing intense market competition. Therefore, it is important for production companies and media platforms to choose the types of movies of good quality to promote and target their audiences, thereby maximising the profit. Training a machine learning model which can help predict the rating level of a movie based on various characteristics such as genre, budget, is significant for this industry. Such a model can not only help viewers to filter out undesirable movies, but it can also provide suggestions for media companies in deciding which genres are likely to receive higher ratings and support for entertainment platforms to improve their recommendation systems. **In this project, the aim is to train a model that is appropriate and accurate for rating category prediction.** The dataset used in this project is derived from the movie database TMDB [1]. The amounts of total entries in the train, evaluate, test, and unlabelled sets are listed in Table 1. The datasets provide both labelled and unlabelled entries, and each entry has rate category and average rate as its class labels. Considering this model focuses on a classification approach rather than regression, the discrete class, rate category will be used as the label.

Train	Evaluate	Test	Unlabelled
100000	20000	20000	254701

Table 1: Count of entries in the data set.

2 Literature Review

Many researchers have devoted their effort to the field of automated movie rating predictions. Lee et al. (2018) highlighted that movie rating predictions can help provide strategies for studios on advertisement, reduce the uncertainty of high-risk investment, thus saving costs. Moreover, they emphasised that ensemble learning methods can effectively improve the performance of a prediction model. In feature selection, which is one of the most important processes in training a model, Ahmad et al.(2020) demonstrate that some common factors such as cast, director, and number of screen releases will highly impact the final result, but these factors have different effects across different regions. Sadashiv et al.(2021) used a variety of machine learning algorithms and compared the performance in their study. Random forest model performed best among all models with an accuracy of 85.20% in the end.

3 Methods

To address the goal, several machine learning algorithms are tested after data preprocessing. Following initial model trials, unimportant features are eliminated based on relevance to improve computational efficiency and model performance. After that, unlabelled data is introduced to do semi-supervised learning experiments to check in this scenario, whether unlabelled data can help further improve the predicting accuracy. Finally, cross validation is used to evaluate whether the model is robust. If the mode is not appropriate-fitting, adjustment of hyper parameters will be performed to optimise the model.

3.1 Data Pre-processing

The dataset features are divided into 37 non-text features and 4 sets of text features that have been processed using bag-of-words (BOW)

and TF-IDF (Term Frequency-Inverse Document Frequency). Among all the characteristics, the "original language" requires preprocessing. All "original language" columns from the 4 datasets are concatenated into a single list to cover all possible languages, and then converted to numerical data. This step ensures that all languages are recognized and consistently encoded which makes this feature suitable for machine learning.

3.2 Baseline

Before starting training any models, ZeroR was chosen to be the baseline for subsequent model evaluation. ZeroR simply predicts the labels based on the majority class in the training set without considering anything else. **The final ZeroR model achieved an accuracy of 0.271 on the evaluate set.** Given that there are 6 categories, this measure suggests the distribution in the dataset is not particularly uniform but also not dominated by a single class. Therefore, this benchmark is acceptable.

3.3 Model Selection

First of all, five sets of features were used to observe behaviours of different models:

- All non-text features
- BOW text features only
- TF-IDF text features only
- All features including BOW text
- All features including TF-IDF text

These were put into training using five common models:

- K Nearest Neighbours(KNN): a model that predicts a new sample using the majority label of its K nearest neighbours.
- Multinomial Naive Bayes(MNB): a probabilistic classifier based on Bayes' theorem, assuming independence of all features.
- Decision Tree(DT): a tree structure that predicts a new sample by learning simple decision rules inferred from the training set.
- Logistic Regression(LR): a statistical model which uses a logistic function to model a binary dependent variable.
- Multi-Layer Perceptron(MLP): an artificial neural network that consists of an input layer, at least one hidden layer, and an output layer, and uses backpropagation for training.

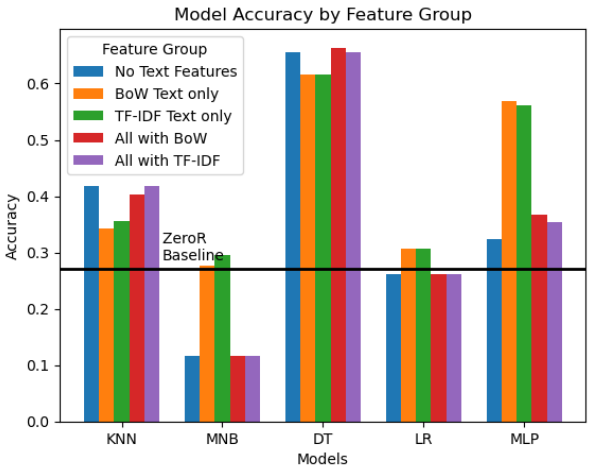

Figure 1: The behaviours of 5 models on 5 different feature sets

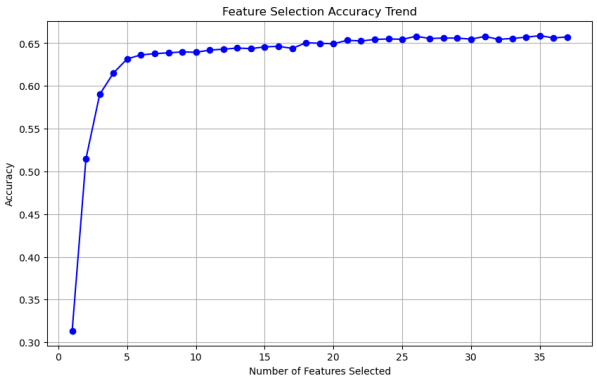

Figure 2: The behaviours of 5 models on 5 different feature sets

Figure 1 is the clustered bar chart depicting model accuracy by feature group. It shows that MNB and LR even struggle to reach the ZeroR baseline, indicating their limitation for this particular task. Though KNN and MLP also do not behave very well, they can occasionally be used to do some auxiliary validation work in the following steps. Across different sets of features, it is evident that **DT stands out and will thus be the main focus** for further training and optimization in this project. The upgraded version of DT, **random forest(RF)**, which creates an ensemble of trees and averages their outputs to improve accuracy, will be introduced to training later.

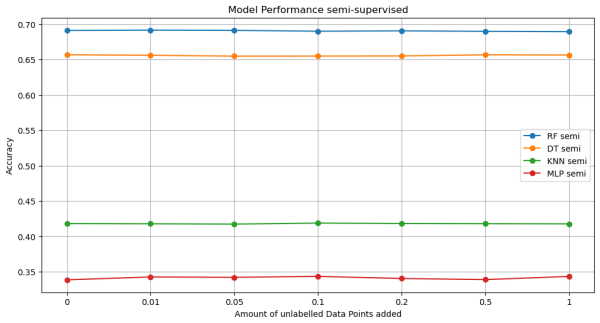

Figure 3: The behaviours of 5 models on 5 different feature sets

3.4 Feature Selection

The ultimate goal of this step is to identify the most important features for predicting movie ratings, thus optimising the performance by eliminating irrelevant features or noise. As shown in Figure 1, while the inclusion of text features may offer some depth to the analysis, it does not lead to a notable improvement on prediction accuracy. Moreover, during the experiments, it is found that the training process with text features is quite time-consuming. Given this low cost performance, to enhance the experiment efficiency, text features will not be a focus in the subsequent research especially when training a complicated model, but they may sometimes be used to do support work. To select the most effective features in the 37 non-text ones, the Recursive Feature Elimination (RFE) method is used to evaluate the relevance of each feature by training a decision tree classifier with varying numbers of features from 1 to 37, and assessing their performance. Figure 2 illustrates the accuracy trend of RFE results. It can be easily inferred that the initial features added significantly contribute to the model's performance. After about 10 features, the accuracy curve begins to flatten, showing only tiny improvements as more features are included. As the research topic pays attention to accuracy, the 35 features which achieve the highest score 0.6589 will be kept for follow-up model training. **Two excluded features are 'product-of-India' and 'product-of-Japan'**, which are almost consistently excluded in other RFE trials based on observations.

3.5 Semi-supervised Learning

Semi-supervised learning leverages both labelled and unlabeled data to improve model performance as sometimes labelled data can be expensive or scarce to obtain. Self training is one of semi-supervised learning techniques. It trains a model on the original labelled dataset and then uses this model to predict labels for the unlabeled data, then the most confident predictions are added to the training dataset iteratively. Figure 3 displays the accuracy of four models undergoing self-training, where varying amounts of unlabeled data were added to the original training set. From the graph, the trend is remarkably stable, indicating that introducing **unlabeled data has minimal impact** on enhancing original performance.

3.6 Evaluation and Hyper-parameter Adjustment

Table 2: Classification Report

Class	Precision	Recall	F1-Score	Support
0	0.697	0.671	0.684	2271
1	0.628	0.705	0.664	2518
2	0.714	0.653	0.682	5593
3	0.746	0.709	0.727	5709
4	0.624	0.758	0.684	2299
5	0.670	0.689	0.680	1610
Accuracy			0.692	20000
Macro Avg	0.680	0.697	0.687	20000
Weighted Avg	0.696	0.692	0.693	20000

Table 2 is the classification report of the performance of a random forest which is trained on the train set and tested on the evaluate set. The overall accuracy stands at 69.2%. Across different rating categories, the report shows varying levels of precision and recall. For example, class 4 shows the highest recall (0.758) indicating that the model is quite good at identifying most of the true instances with this label. However, its precision is relatively lower (0.624), which suggests that there are a number of instances where the model predicted as class 4 but was incorrect. With fewer supports, the model still performs well on the extreme classes (0 and 5) in terms of F1-score, which balances the precision and recall. This indicates that features distinguishing high or low ratings are well-captured by the model. To further improve the performance, GridSearchCV was used to do hyperparameter adjustment. This method builds and evaluates a model for each combination of hyperparameters. In this experiment, a wide variety of possible parameters are assigned to the parameter grid, leading to nearly one thousand of parameter combinations in total which require numerous fit processes. After ten hours of exploration, GridSearchCV ultimately identi-

fied a set of optimal hyperparameters. However, the result is disappointing after trying the tuned model. The prediction accuracy even dropped from 69%, achieved by the model with default parameters, to below 60%. Due to this outcome, it can be concluded that a **Random Forest without any specified tuning parameters actually performs the best.**

4 Results

In this research, various machine learning models are assessed to predict movie ratings using the TMDB dataset. The Random Forest with default settings, significantly outperformed other models, achieving an overall accuracy of nearly 70% on the evaluation set. The feature selection process identified 35 critical non-text features that slightly enhanced model performance, while text features were not further investigated due to the low impact and low time cost performance. Although semi-supervised learning was applied by using self-training with unlabeled data, they did not substantially improve the accuracy and were time-consuming at the same time. Finally, hyperparameter tuning via GridSearchCV did not yield a better result, indicating that the RF model with default parameters already balances complexity and performance very well. This highlights the effectiveness of RF in handling such complicated tasks.

5 Discussion/Critical Analysis

5.1 Factors of the Outstanding Accuracy of DT & RF over Others

Decision trees are suitable for handling non-linear problems and capturing the relationships between complex data. The upgraded version, Random Forests, by constructing multiple decision trees and counting their votes, achieves better generalisation than individuals. In comparison, KNN is sensitive to high-dimensional data, which can lead to a disaster. MNB assumes independence among features but this assumption is invalid in this research scenario; this is the main reason for the poor performance. Similarly, LR has limited capability to catch non-linear correlations between features and may need complex extensions for multi-class problems. At last, MLP is powerful, but it requires sufficient training data and is extremely sensitive to hyperparameters.

5.2 Explanations for Minimal Gain from Unlabelled Data

The aim of introducing unlabeled data primarily is to increase the volume of the training set. However, if the model has already achieved an excellent fitting and reached its performance ceiling by training with the labelled data, adding extra sources might not lead to further enhancement. Moreover, the unlabeled data may not contain sufficient novelty, or may be similar in distribution as the training set. Then the impact of this data on improving the model's generalisation ability is limited, since most of the unlabeled data is used to confirm the already known decision boundaries rather than explore new knowledge.

5.3 Analysis for Performance Decline after Hyperparameter Tuning

During the hyperparameter tuning process, it is possible to increase the regularisation of the model unexpectedly. Parameters such as min samples split and min samples leaf can prevent the model from over fitting but can also prevent the model from learning enough information from the original data, thus reducing the final performance score. Additionally, one point worth mentioning is that the model trained on the train set, when used to predict the same train set, there is a drop in accuracy from 99% to 79%. This sharp decline indicates that the new parameter settings have oversimplified the model or imposed too many constraints, preventing it from capturing key patterns and relationships in the data. Consequently, this set of hyperparameters weakens the model's generalisation ability.

6 Conclusion

This research aimed to develop a powerful machine learning model to predict movie ratings using the TMDB dataset. Finally, **an accuracy of nearly 70% was achieved by Random Forest.** Experiments have shown that including text features, semi-supervised learning with unlabelled data, and hyperparameter tuning via GridSearchCV did not substantially improve the results. These findings imply that the non-text features and the default setting of parameters are already optimal for training a good model for movie rating predictions.

References

Ahmad, I. S., Bakar, A. A., Yaakub, M. R., & Muhammad, S. H. (2020). A survey on machine learning techniques in movie revenue prediction. *SN Computer Science*, *1*(4), 235.

Lee, K., Park, J., Kim, I., & Choi, Y. (2018). Predicting movie success with machine learning techniques: ways to improve accuracy. *Information Systems Frontiers*, *20*, 577–588.

Sadashiv, S., Sween, S., & Sankruth, S. (2021). Movie success prediction using machine learning. *Int. Res. J. Mod. Eng. Technol. Sci*, *3*, 2021–2024.