COMP30027 Project 2 Report

Anonymous
Individual Project

1. Introduction

Watching movies is one of the most popular hobbies worldwide. Both critics and audiences contribute ratings for movies on platforms like IMDB. To analyze the factors influencing these movie ratings, this project aims to build models to predict movie ratings using the IMDB 5000 Movie Dataset (Carol Z., 2018) with supervised machine learning methods.

In this report, details of methodology, evaluation on model results and errors will be discussed.

2. Methodology

Before building the prediction models, data was pre-processed and features were carefully selected for evaluation.

2.1 Data Pre-processing

Data pre-processing was performed based on the data types before selecting suitable features for evaluation.

2.1.1 MinMax Scaling

The IMDB dataset includes features with continuous data, such as movie duration, Facebook likes of actors and directors, and gross revenue. Since these features have different data ranges, Min-Max normalization was applied to prevent bias from features with larger magnitudes and to reduce the impact of outliers on model performance.

2.1.2 One-hot encoding

Some features in the dataset are categorical, for example, language, content rating, which contain string values and cannot be processed by classifiers. One-hot encoding was applied to these categorical features to provide classifiers with information about each category and to avoid implying any numerical relationships between categories.

For the features actor_1_name, actor_2_name, and actor_3_name, the results of one-hot encoding were combined into a single feature (actor_name) because all three features indicate the presence of a specific actor in the movie.

For the features plot_keywords, director_name, and actor_name, only the top 200, 200, and 400 categories, respectively, were retained. Since these features have numerous categories, dropping some binary columns helped reduce the dimensionality of the encoded feature space. Categories present in fewer than five instances were excluded, as they may not reliably determine movie ratings and could lead to overfitting.

The feature title_year was grouped by decades (e.g., 1980 to 1989) to add meaning to the feature and to help classifiers better explore the apparent effects of year groups on movie ratings compared to individual years.

For the feature movie_title, stopwords such as ‘the’ and ‘is’ were removed, as they do not carry significant meaning or relationship with the movie rating. Removing them reduced noise in the dataset, after which one-hot encoding was applied to the remaining words in the feature.

After one-hot encoding, some duplicate columns were identified, as the same category could be present in different features (e.g., a category in director_name might also be in actor_name). These duplicate columns were combined for simplification.

2.2 Feature Extraction and Selection

After pre-processing the training data, 1126 features were generated. To improve model performance, the χ^2 test was used to select informative features that are useful for predicting movie ratings.

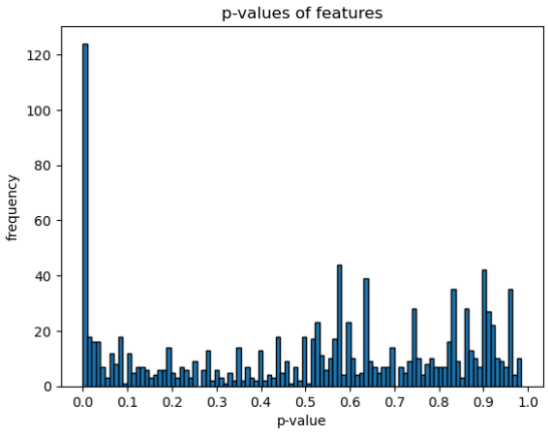

Figure 1- p-values of features before extraction and selection

Figure 1 shows the p-values of all features. The p-value with the highest frequency is the bar at around 0.0 (> 120), representing the most informative features.

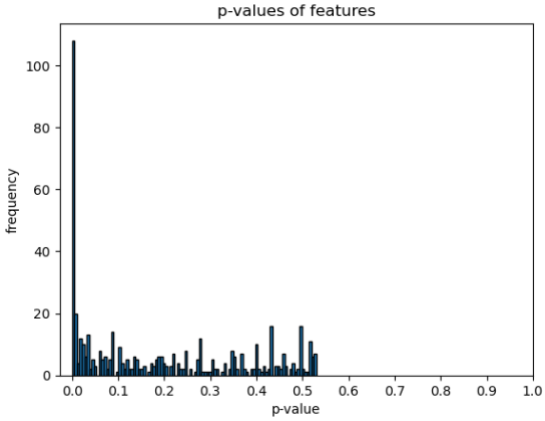

Figure 2- p-values of features after extraction and selection

After careful consideration, 500 of the best features were selected. Figure 2 shows the p-values of these selected features, with most of them being lower than 0.5.

2.3 Learners

Three supervised machine learning models were implemented for this project: Random Forest, Logistic Regression, and Linear SVM. Five-fold cross-validation was conducted for hyperparameter tuning and error analysis of each learner. The training data was divided into five equal folds for cross-validation, which helps prevent bias from certain data in the training dataset.

3. Results

3.1 Random Forest

The Random Forest model was built with 200 decision trees (n_estimators) and tested with different maximum depths:

Max depth	Accuracy
10	0.6748
50	0.7290
100	0.7314
200	0.7314
500	0.7314

Table 1 - Random Forest Model Accuracy with different maximum depth

In Table 1, he accuracy of the model reaches its peak when the maximum depth is 100 and remains the same with greater depths, indicating that overfitting starts to occur when the max depth exceeds 100. Therefore, a maximum depth of 100 was used in the final model.

3.2 Logistic Regression

When building the Logistic Regression model, L2 regularization (Ridge regularization) was used. The model was tested with different C values:

C value	Accuracy
0.25	0.7031
0.5	0.7114
1	0.7104
5	0.7021
10	0.7014
20	0.6944

Table 2 – Logistic Regression Model Accuracy with different C values

In Table 2, the model accuracy is highest when the C value is 0.5, indicating that regularization with a C value greater than 0.5 is too weak and results in overfitting. Therefore, C=0.5 was used in the final model.

3.3 Linear SVM

When building the Linear SVM model, the model was tested with different C values:

C value	Accuracy
0.25	0.6954
0.5	0.6911
1	0.6854
5	0.6768
10	0.6741
20	0.6714

Table 3 – Linear SVM Model Accuracy with different C values

In Table 3, the model accuracy is highest when

the C value is 0.25, indicating that regularization with a C value greater than 0.25 is too weak and results in overfitting. Therefore, C=0.25 was used in the final model.

3.4 Model Performance on Test Data

Model	Best Accuracy
Random Forest	68.617%
Logistic Regression	68.351%
Linear SVM	65.691%

Table 4 – Model Accuracy

Table 4 shows the model performance on 50% of test data. The Random Forest model has slightly higher accuracy than the Logistic Regression model. Random Forest and Logistic Regression models outperformed the Linear SVM model.

4. Discussion

This session evaluates model behaviour and conducts error analysis for each model.

4.1 Random Forest

The random Forest model has the highest accuracy among the models because it can capture non-linear patterns between features and movie ratings. For example, Facebook likes of actors may have a non-linear relationship with movie ratings. Moreover, since Random Forest is an ensemble approach, multiple decision trees are trained using subsets of the dataset, making the model less sensitive to noisy data, with low variance and less chance of bias.

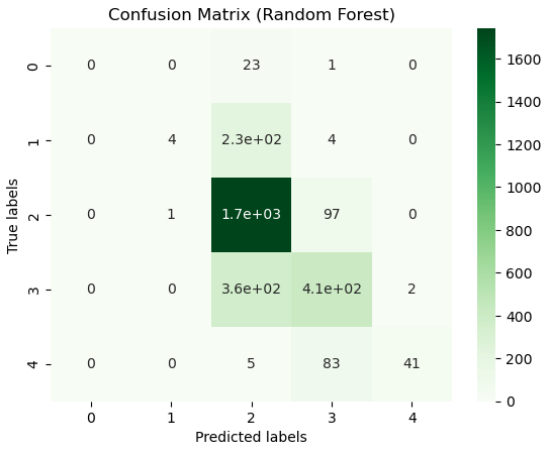

Figure 3 – Random Forest confusion matrix

Figure 3 shows the confusion matrix of the Random Forest model, created using 5-fold cross-validation on the test dataset. Most movies with a rating of 2 are correctly predicted. However, movies with ratings of 0 or 1 tend to be overpredicted, and none of the movies with a rating of 0 are correctly predicted. This is due to the lack of samples with a rating of 0 for training, resulting in high variance, overfitting, and lower accuracy. Movies with ratings of 3 or 4 tend to be under-predicted.

4.2 Logistic Regression

The Logistic Regression model has relatively high accuracy (over 60%) because it evaluates the coefficients of each feature in the dataset, which is useful in interpreting the relative importance of each feature and making more accurate movie rating predictions.

However, it has slightly lower accuracy than the Random Forest model because the Logistic Regression model assumes linear relationships between features and movie ratings, failing to capture complex and non-linear relationships. Additionally, the model is more sensitive to noisy data than the Random Forest model, due to incorrect coefficient
determination.

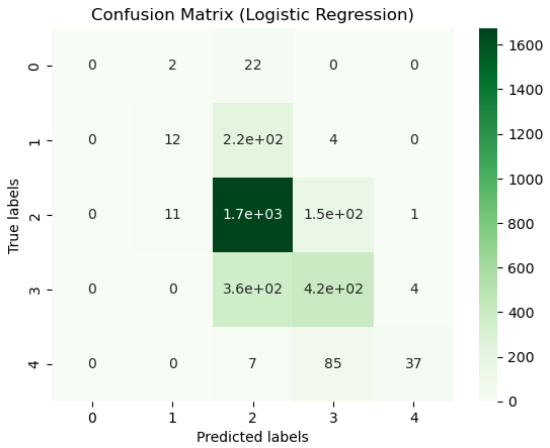

Figure 4 – Logistic Regression confusion matrix

Figure 4 shows the confusion matrix of the Logistic Regression model. Among all class labels, the model accuracy is highest when predicting movies with a rating of 2, although slightly poorer than the Random Forest model. The Logistic Regression model has slightly higher accuracy in predicting labels with fewer instances in the dataset, such as movies with ratings of 1 and 3, due to the regularization techniques applied, reducing the chance of overfitting compared to Random Forest models.

4.3 Linear SVM

Linear SVM model has the poorest performance among the three models because the movie rating dataset is high-dimensional, with 500 features. The Linear SVM model may struggle to determine an optimal hyperplane that effectively separates the movie rating labels.

Additionally, similar to logistic regression, Linear SVM assumes that class labels are separated by linear boundaries, failing to capture non-linear relationships between features and labels.

Moreover, the Linear SVM model assumes equal importance for all features. As a result, the model performance was hindered because the influence of important features on the class label was weakened by the presence of less significant features.

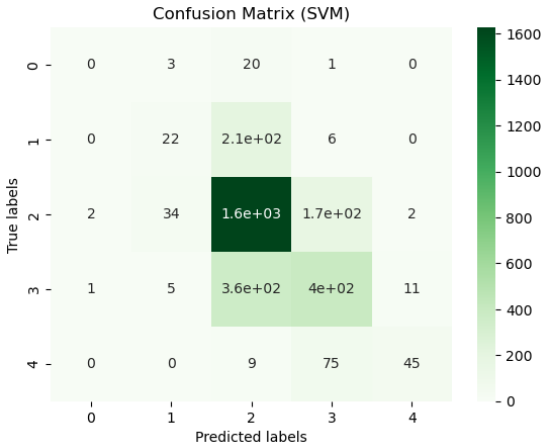

Figure 5 – Linear SVM confusion matrix

Figure 5 shows the confusion matrix of the Linear SVM model. Among all class labels, the model accuracy is highest when predicting movies with a rating of 2. However, the model accuracies for predicting movies with ratings of 2 and 3 are lower than those of the other two models. The model performs slightly better than the other two models when predicting movies with ratings of 1 and 4.

5. Conclusions

In conclusion, Random Forest, Logistic Regression, and Linear SVM models were built to predict movie ratings using the IMDb movie dataset. All three models perform well in predicting movie ratings, with accuracy greater than 60%. The Random Forest model performed the best among the three models. With more computational resources in the future, additional informative features, such as natural language processing on movie user reviews, could be explored. Additionally, ensemble methods could be applied to Logistic Regression and Linear SVM models to improve model performance.

6. References

Carol Z.2018. *IMDB 5000 Movie Dataset*. Retrieved from https://www.kaggle.com/datasets/carolzhangdc/imdb-5000-movie-dataset

(Word count: 1404)