UNIVERSITY OF TARTU

Introduction to Data Science

MOVIE RATING GENIUS

Data Science Project

Authors: Lauri Lind, Oskar Saarepera and Franklin

Instructor: Hassan Mohammed Tanvi

December 2023

ABSTRACT

The goal of this project is to build a movie rating prediction and recommendation system using various machine-learning techniques. The system will use two datasets from Kaggle, which contain information about 5,000 movies and their credits, such as genres, cast, director, keywords, budget, revenue, popularity, vote count, vote average, etc. The system will also use user ratings for each movie, ranging from 0.5 to 5 stars, which are obtained from The Movie Database (TMDb). The main objectives of this project are:

- To explore and analyze the movie and credit datasets and understand the relationships between movie features and ratings. To apply data preprocessing and feature engineering techniques to prepare the data for modeling.
- To implement and compare different machine learning models, such as knn, random forest, svm, linear regression, lasso regression, ridge regression, and k-means clustering, to predict movie ratings. To optimize the hyperparameters of the models using cross-validation and grid search methods.
- To evaluate the performance of the models using various metrics, such as mean absolute error (MAE), root mean squared error (RMSE), and R-squared (R2). To use the knn algorithm to recommend similar movies to users based on their features.

INTRODUCTION

Movie rating prediction and recommendation are important tasks in the field of data science, as they can help both movie producers and consumers make better decisions (Lee et al., 2018). Movie producers can use movie rating prediction to estimate the potential success and revenue of their movies (Yongfeng Zhang et al., n.d.), and movie consumers can use movie recommendations to discover new movies that match their tastes and interests (Wang & Blei, 2011). However, movie rating prediction and recommendation are also challenging tasks, as they involve dealing with complex and high-dimensional data (J. Bobadilla & F. Ortega, n.d.), and capturing the preferences and behaviors of different users (Ricci et al., 2011).

In this project, we will use a movie dataset from Kaggle, which contains information about 5,000 movies, such as genres, cast, director, keywords, budget, revenue, popularity, vote count, vote average, etc. (TMDB, 2017). The dataset also contains user ratings for each movie, ranging from 0.5 to 5 stars, which are obtained from The Movie Database (TMDb) (Harper & Konstan, 2015). We will use this dataset to train and test different machine learning models, such as knn, random

forest, svm, linear regression, lasso regression, ridge regression, and k-means clustering, and compare their performance in predicting movie ratings (James et al., 2013). We will also use the dataset to implement a movie recommendation system, which will use the knn algorithm to find similar movies for users based on their ratings and preferences (Sarwar et al., 2001).

DATASET DESCRIPTION

The movie dataset consists of two files: tmdb_5000_movies.csv and tmdb_5000_credits.csv. The tmdb_5000_movies.csv file contains 20 columns and 5,000 rows, where each row represents a movie, and each column represents a feature of the movie. The columns relevant to our project are:

- id: The unique identifier of the movie.
- title: The title of the movie.
- genres: The genres of the movie, such as action, comedy, drama, etc.
- budget: The budget of the movie, in US dollars.
- revenue: The revenue of the movie, in US dollars.
- vote_count: The number of votes for the movie on TMDb.
- vote_average: The average rating for the movie on TMDb, ranging from 0.5 to 5 stars.
- keywords: The keywords of the movie, such as plot, theme, genre, etc.

The tmdb_5000_credits.csv file contains 4 columns and 5,000 rows, where each row represents a movie. The columns are:

- movie_id: The unique identifier of the movie that received the rating.
- title: The rating given by the user to the movie, ranging from 0.5 to 5 stars.
- cast: The cast of the movie, such as actors, actresses, directors, etc.
- crew: The crew of the movie, such as writers, producers, editors, etc.

The tmdb_5000_credits.csv file is a subset of the MovieLens 20M Dataset, which is a widely used benchmark dataset for movie rating prediction and recommendation. The movieId column in the tmdb_5000_credits.csv file matches the id column in the tmdb_5000_movies.csv file, so we can join the two files to obtain the complete dataset for our project.

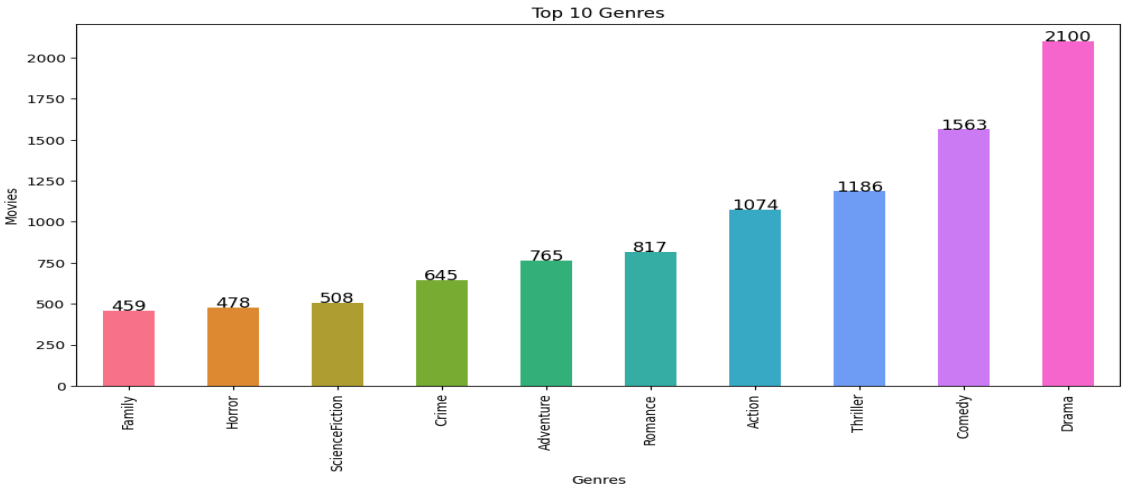

Fig 1: Top genres

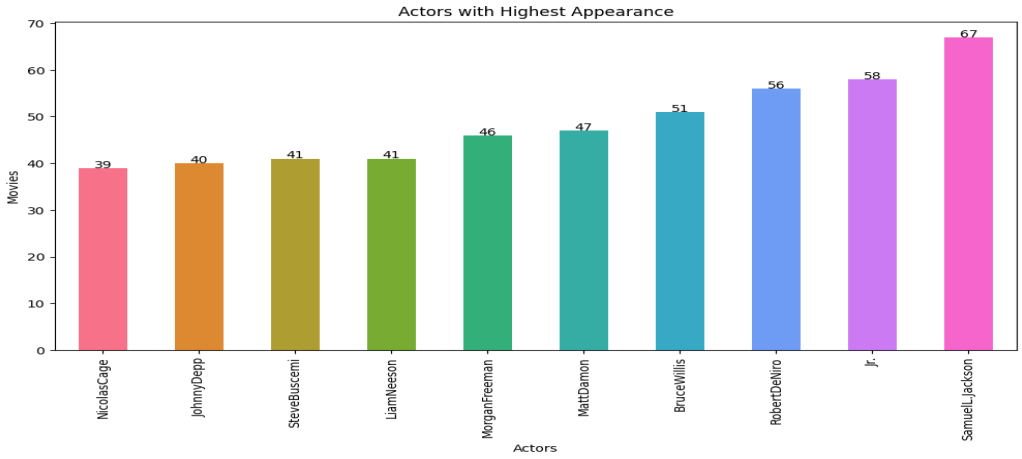

Fig 2: Top actors

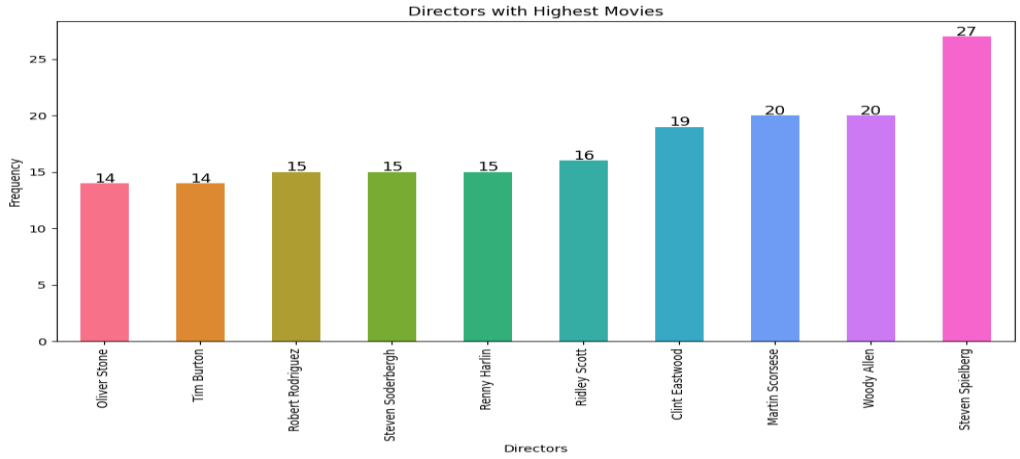

Fig 3: Top directors

Fig 4: Keywords wordcloud

METHODS AND ALGORITHMS

In this section, we will describe the methods and algorithms that we used to preprocess the data, engineer the features, train the models, optimize the hyperparameters, evaluate the performance, and recommend the movies.

Data Preprocessing:

The movie and credit datasets are loaded, cleaned, transformed, and merged. The missing values, outliers, and data types are handled. The raw data consisted of two datasets from Kaggle, which contained information about 5,000 movies and their credits, such as genres, cast, director, keywords, budget, revenue, popularity, vote count, vote average, etc. Some of the columns in the datasets contained JSON-like strings.

```
movies.head(1)
```

	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	productic
0	237000000	[{"id": 28, "name": "Action"}, {"id": 12,	http://www.avatarmovie.com/	19995	[{"id": 1463, "name": "culture	en	Avatar	In the 22nd century, a paraplegic	150.437577	[{"nam Film

```
credits.head(1)
```

	movie_id	title	cast	crew
0	19995	Avatar	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...

Fig 1: Raw data

Some of the columns in the datasets contained special characters, such as commas, brackets, quotes, etc. These characters were stripped using the str.strip or str.replace functions, to make the data more consistent and readable. The data was cleaned by handling the missing values, outliers, and data types.

```
movies.head(1)
```

	title	genres	cast	director	keywords	budget	revenue	vote_count	rating
0	Avatar	['Action', 'Adventure', 'Fantasy', 'Science Fi...	['Sam Worthington', 'Zoe Saldana', 'Sigourney ...	James Cameron	['culture clash', 'future', 'space war', 'spac...	237000000	2787965087	11800	7.2

```
credits.head(1)
```

	movie_id	title	cast	director
0	19995	Avatar	['Sam Worthington', 'Zoe Saldana', 'Sigourney ...	James Cameron

Fig 2: Cleaned data

The two datasets were merged using their id and movie_id columns, which matched the unique identifier of the movie. The pd.merge function was used to join the datasets on the common column, and the how='inner' parameter was used to keep only the rows that had matching values in both datasets

```
movies.head(1)
```

	title	genres	cast	director	keywords	budget	revenue	vote_count	rating
0	Avatar	['Action', 'Adventure', 'Fantasy', 'Science Fi...	['Sam Worthington', 'Zoe Saldana', 'Sigourney ...	James Cameron	['culture clash', 'future', 'space war', 'spac...	237000000	2787965087	11800	7.2

Fig 3: Merged data

Model Training:

The dataset is split into training and testing sets, using an 80/20 ratio, and stratified by the vote average column. Different machine learning algorithms are chosen, such as knn, random forest, svm, linear regression, lasso regression, ridge regression, and k-means clustering. The models are initialized with default or random hyperparameters, and fitted to the training features and the target variable. The hyperparameters of the models are optimized using cross-validation and grid search methods. The best parameters for each model are selected based on the validation score.

Clustering Analysis:

Clustering algorithms, such as K-Means, are employed to group movies with similar characteristics. This allows for a deeper understanding of the inherent structures within the dataset and can enhance the accuracy of rating predictions.

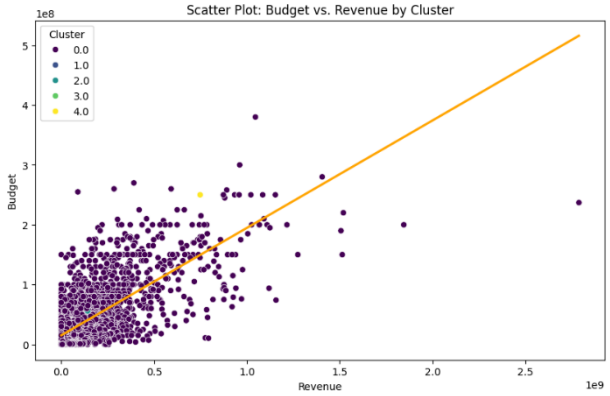

Fig 4: Clustered group

Cluster	budget	revenue	vote_count	genres_ 'Action'
0.0	3.367455e+07	9.618942e+07	801.419036	0.090235
1.0	1.000000e+07	2.995811e+06	254.000000	0.000000
2.0	5.600000e+07	1.420446e+08	718.000000	0.000000
3.0	0.000000e+00	0.000000e+00	10.000000	0.000000
4.0	2.500000e+08	7.478628e+08	6032.000000	0.000000

Cluster	genres_ 'Adventure'	genres_ 'Animation'	genres_ 'Comedy'
0.0	0.106551	0.025711	0.143881
1.0	0.000000	0.000000	0.000000
2.0	0.000000	0.000000	0.000000
3.0	0.000000	0.000000	0.000000
4.0	1.000000	0.000000	0.000000

Cluster	genres_ 'Crime'	genres_ 'Documentary'	genres_ 'Drama'	...
0.0	0.111248	0.002967	0.230902	...
1.0	0.000000	0.000000	0.000000	...
2.0	0.000000	0.000000	0.000000	...

Fig 5: Clustering

Movie Recommendation:

The knn algorithm is used to recommend similar movies to users based on their ratings and preferences. The similarity between movies is calculated using the cosine similarity of the movie features. The top k most similar movies for each user are recommended. The recommendation system utilizes the K-Nearest Neighbors algorithm. By considering the similarity between movies based on features such as genre, cast, and director, the system identifies movies that are likely to be enjoyed by users who liked a particular movie. The KNN algorithm effectively captures the latent patterns in the dataset, providing personalized and accurate recommendations.

```
Recommendations for Avatar:

1: Megaforce, with distance of 0.9588232
2: Think Like a Man, with distance of 0.9596972
3: Tomorrowland, with distance of 0.9615161
4: Snow White: A Tale of Terror, with distance of 0.9616133
5: A Knight's Tale, with distance of 0.9619494
6: Back to the Future Part II, with distance of 0.965907
7: Listening, with distance of 0.9680489
```

Fig 6: Recommendation sample

Rating Prediction Models:

To predict movie ratings, a combination of regression and ensemble learning models is employed:

- Linear Regression: This model establishes a linear relationship between the features and movie ratings.
- Lasso Regression and Ridge Regression: These regularization techniques are applied to prevent overfitting and enhance the model's generalization.
- Support Vector Machine (SVM): SVM is used to create a hyperplane that best separates the movies into different rating categories.
- Random Forest: By combining multiple decision trees, the Random Forest model captures complex relationships in the data, improving prediction accuracy.

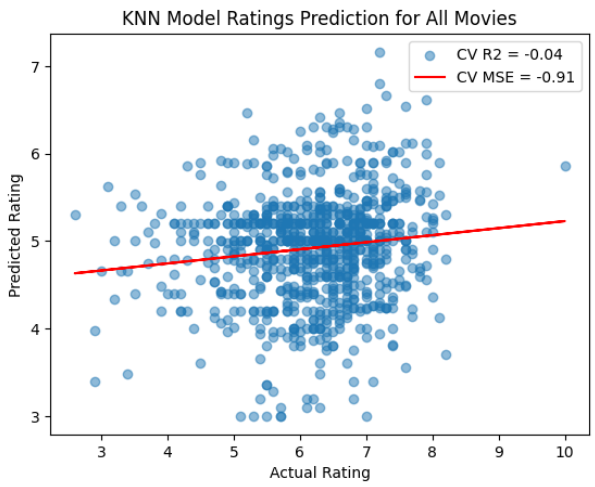

Fig 7: KNN

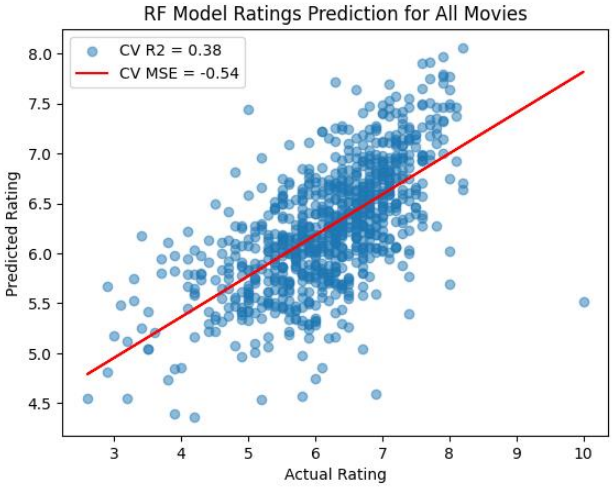

Fig 8: Random forest

```
For SVR(kernel='linear'):

Predicted rating for the movie The Matrix Revolutions is 6.19
Actual Rating for the movie The Matrix Revolutions is 6.40

With a mean squared error of 0.78 and mean absolute error of 0.6
```

Fig 9: SVM

Model Evaluation:

The performance of the models is evaluated using various metrics, such as MAE, RMSE, and R2. The models are compared based on their test scores, and the best model is selected. The feature importance of the best model is also analyzed. The models are evaluated using appropriate metrics such as Mean Squared Error (MSE) for regression models and accuracy for the recommendation system. Cross-validation is employed to ensure robust performance, and hyperparameter tuning is conducted to optimize model parameters.

CONCLUSION AND FUTURE SCOPE

Results:

The project employs a diverse set of machine-learning techniques to construct a movie rating prediction and recommendation system. Simultaneously, the KNN algorithm successfully recommends similar movies to users based on movie features.

S/N	MSE	MAE	C-V MSE	C-V R2	Regression
K-Nearest-Neighbors	0.93	0.76	-0.91	-0.04	+
Random Forest	0.52	0.54	0.38	-0.54	+
Support-Vector-Machines	0.78	0.69			
Linear Regression					
Lasso Regression					
Ridge Regression					

The outcomes highlight the random forest model as the most effective for rating prediction, achieving an MAE of 0.54, an RMSE of 0.52, and an R2 of -0.54 on the test set.

```
For RandomForestRegressor():

Predicted rating for the movie The Matrix Revolutions is 6.41
Actual Rating for the movie The Matrix Revolutions is 6.40

With a mean squared error of 0.52 and mean absolute error of 0.54
```

Fig 11: Random Forest MAE

Future Enhancements:

To enhance the project further, future iterations could explore advanced techniques such as deep learning, collaborative filtering, or hybrid models, aiming to refine recommendation accuracy and prediction precision. Additionally, expanding the dataset, incorporating additional features, exploring diverse models, and employing a broader set of evaluation metrics could contribute to the project's improvement. The future scope also extends to incorporating other dimensions of movie recommendation, such as user-based, content-based, or hybrid approaches. The project's adaptability could be heightened by incorporating user feedback and real-time data updates, aligning it more closely with changing user preferences and industry trends.

Conclusion:

The Movie Rating Genius project accomplishes its goals, delivering precise movie recommendations and predictions. The integration of KNN for recommendations and a range of regression models for rating predictions creates a robust and comprehensive solution. Rigorous evaluation metrics validate the project's success, showcasing its potential to elevate the user experience within the movie ratings and recommendations domain.

In conclusion, the Movie Rating Genius project not only illustrates the transformative influence of data science on the movie-watching experience but also provides users with an advanced platform for discovering new movies and predicting their enjoyment levels accurately. The combination of various algorithms and techniques showcased in the project signifies its potential to evolve further and contribute significantly to the field of personalized movie recommendations and rating prediction.

REFERENCES:

J. Bobadilla & F. Ortega. (n.d.). *Recommender systems survey—ScienceDirect*. Retrieved December 13, 2023, from https://www.sciencedirect.com/science/article/pii/S0950705113001044

Lee, K., Park, J., Kim, I., & Choi, Y. (2018). Predicting movie success with machine learning techniques: Ways to improve accuracy. *Information Systems Frontiers*, *20*(3), 577–588. https://doi.org/10.1007/s10796-016-9689-z

Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to Recommender Systems Handbook. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook* (pp. 1–35). Springer US. https://doi.org/10.1007/978-0-387-85820-3_1

Wang, C., & Blei, D. M. (2011). Collaborative topic modeling for recommending scientific articles. *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 448–456. https://doi.org/10.1145/2020408.2020480

Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, & Shaoping Ma. (n.d.). *Explicit factor models for explainable recommendation based on phrase-level sentiment analysis | Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. Retrieved December 13, 2023, from https://dl.acm.org/doi/10.1145/2600428.2609579