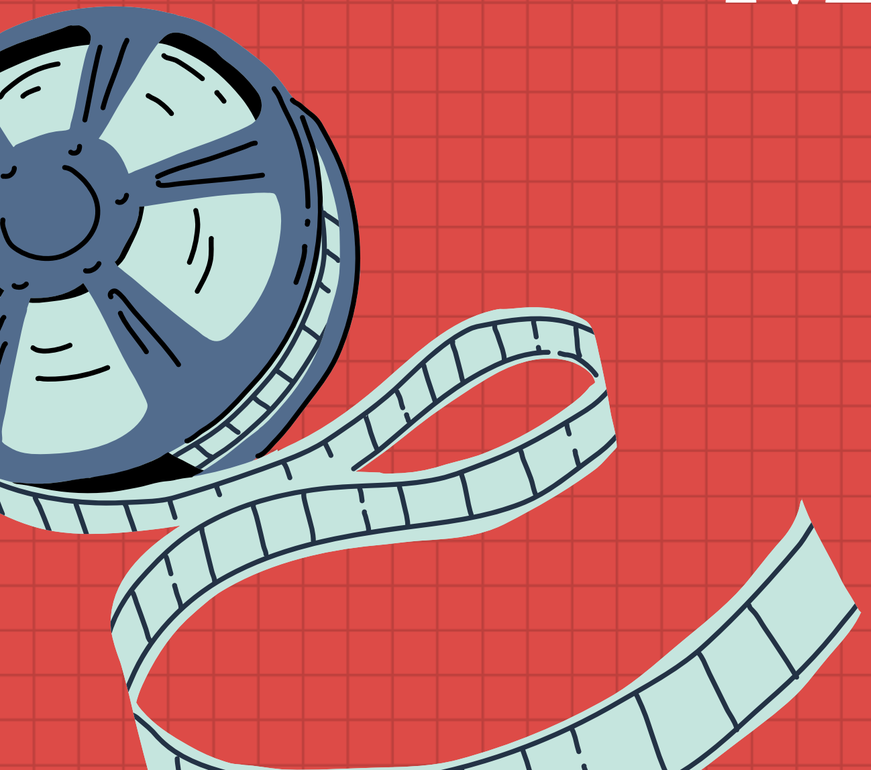
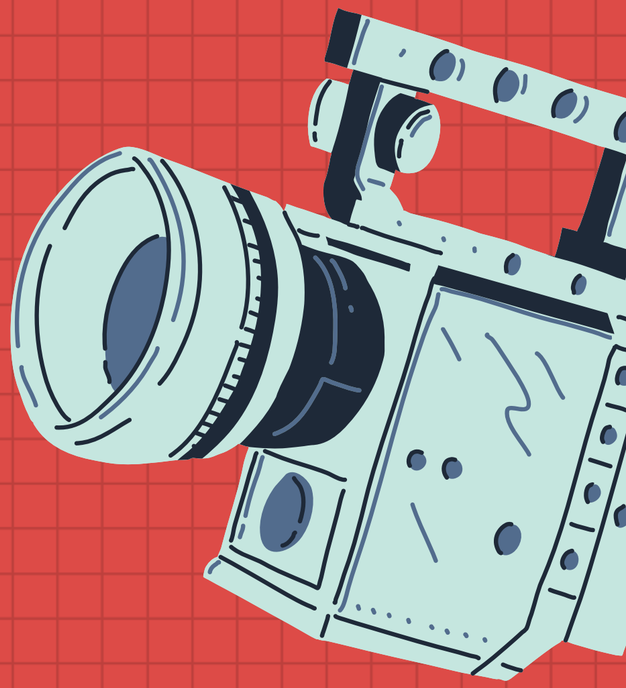
Intro to Data Science

Movie ratings

Final Project





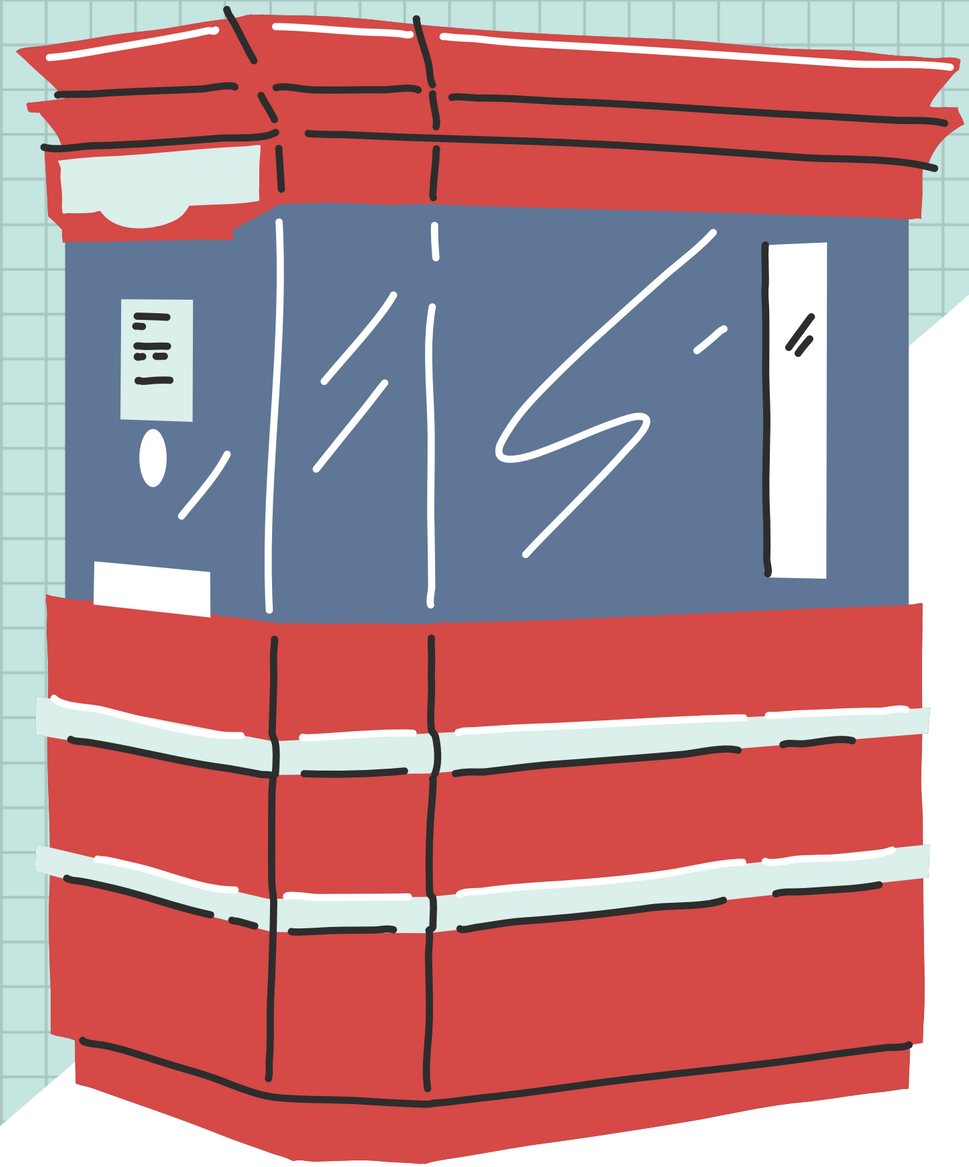
Github & schedule

Github:
https://github.com/Khoa-Nguyen-Kevin/hcmus_nmkhdl_project/tree/Khoa

Schedule:

			WEEK 1							WEEK 2							WEEK 3							WEEK 4						
PHASE	TASK	MEMBER	16/11	17/11	18/11	19/11	20/11	21/11	22/11	23/11	24/11	25/11	26/11	27/11	28/11	29/11	30/11	01/12	02/12	03/12	04/12	05/12	06/12	07/12	08/12	09/12	10/12	11/12	12/12	13/12
Data Collection	Planning Meeting	All																												
	Writing code	Quang + Vũ																												
	Format and Write notebooks	Quang + Vũ																												
	Collecting Data	Quang + Vũ																												
	Draw timeline + reports	Khoa																												
	Research topic/data	Thanh																												
Data Preprocessing + Exploration	Preprocessing	Khoa + Thanh																												
	Making Questions + Answering	Khoa + Thanh + Vũ: 1Q; Quang: 2Q																												
	Format and Write notebooks	All																												
Data Modeling	Planning meeting (Decide	All																												
	Train + Test Model	Khoa + Thanh + Vũ																												
	Comparison and report	Khoa																												
	Format and Write notebooks	Khoa + Thanh + Vũ																												
Others	Create presentation slides	Quang																												

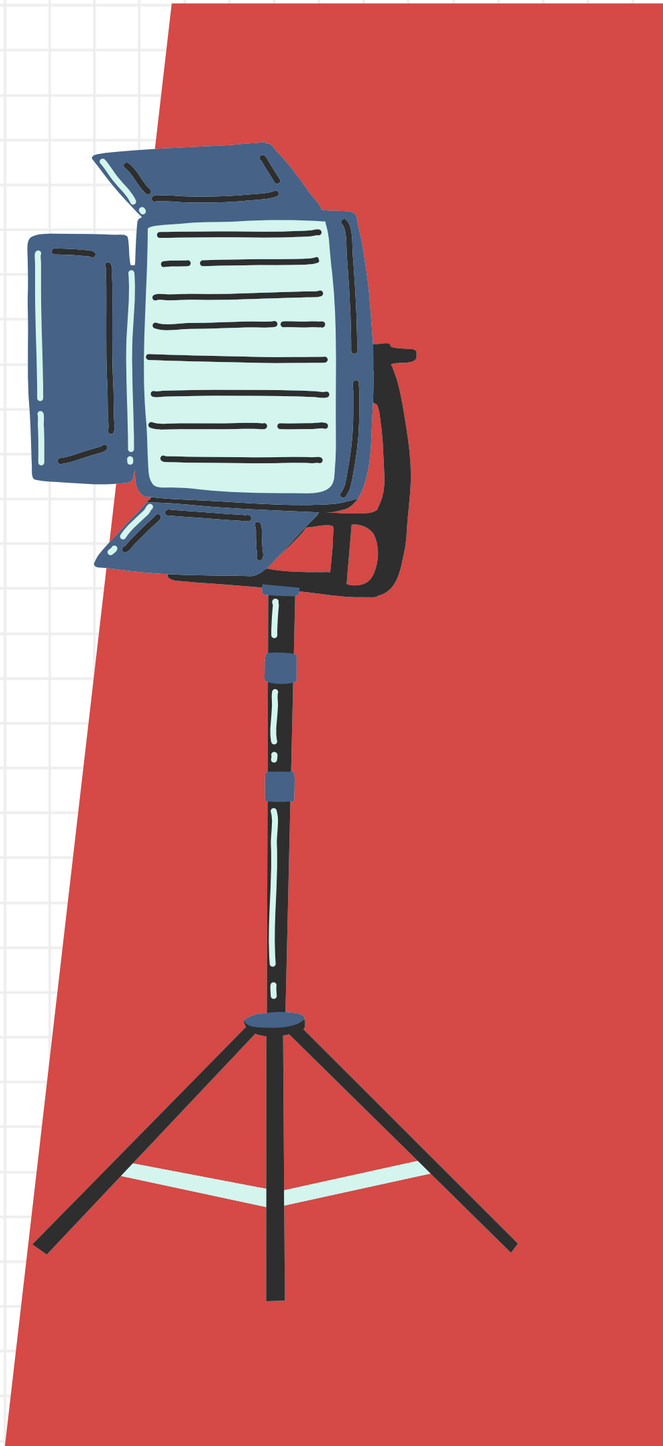
Contents



1. Topic Introduction

We are curious about how movies from various type of genres, classification,... have their ratings affected

- Collect data from RottenTomatoes/IMDB

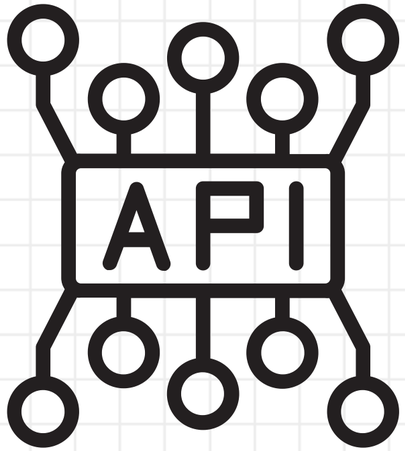


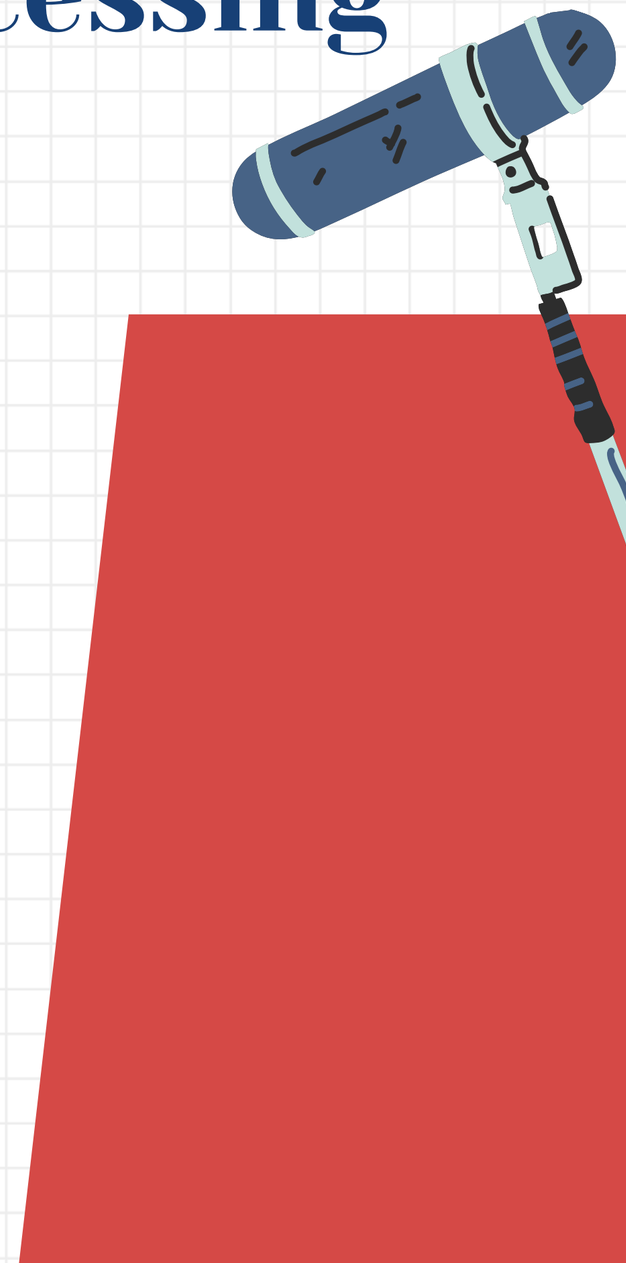
2. Data Collection and Preprocessing

Collect data through API

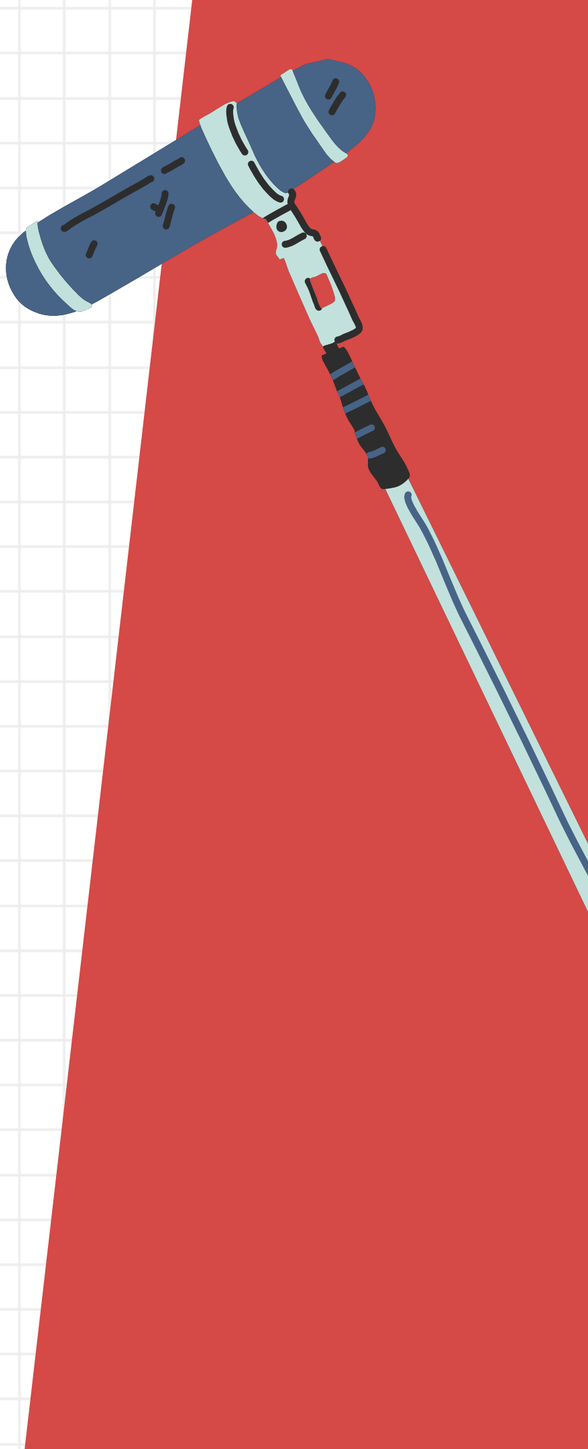
IMDb

- Find pattern of API to get API
- Check the api with postman to see the params and headers
- Check the api for others genre for the rule between api changes
- Send requests and parse jason





Collect data through HTML parsing

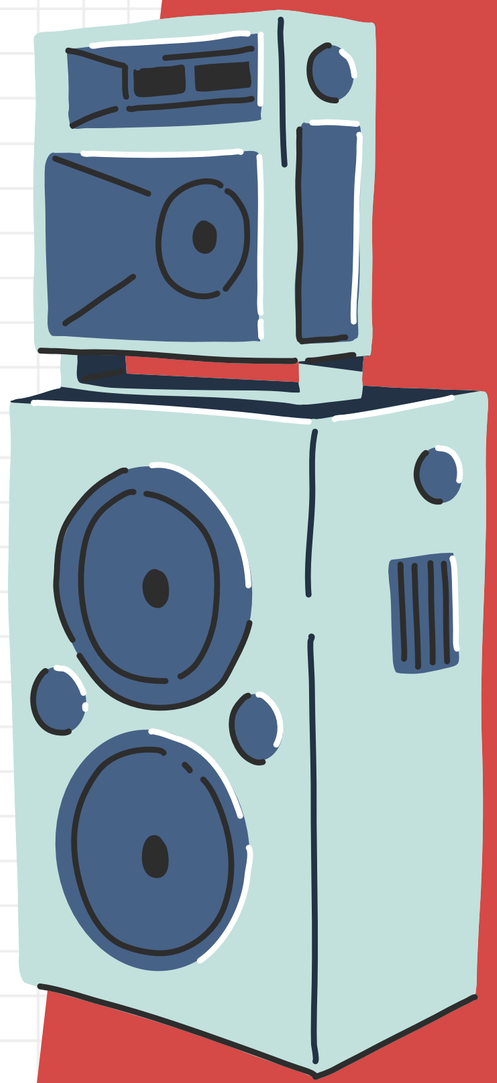


- Send requests to get HTML elements
- Collect movie links by genres
- Parse HTML elements
- Collect data from each movies, save links for later avoid duplicated crawl
- Some elements use Shadow-DOM object -> Use Selenium

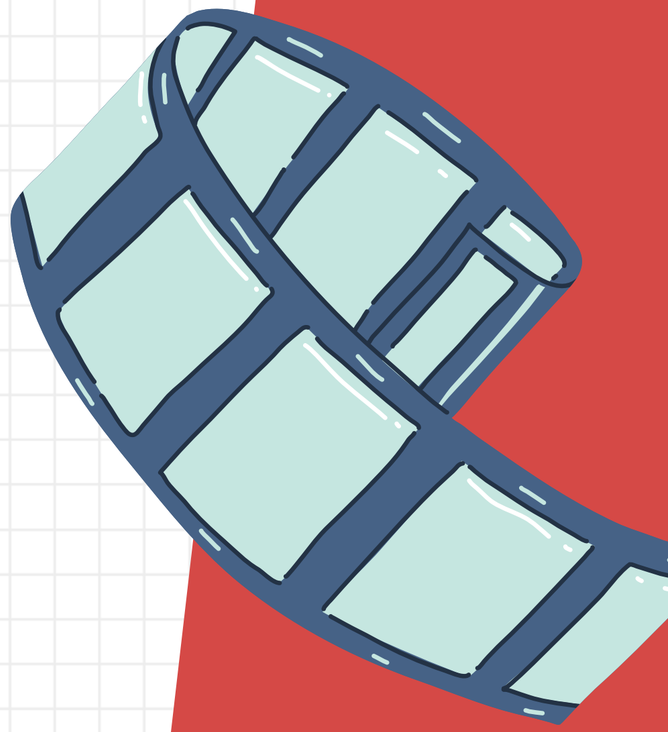
Data set

Our dataset collected dataset has:

- **1216** rows
- **10** fields:
  - Name: movie's name
  - Genre: movie's main genre
  - Tomatometer Score: rating score by critics
  - Tomatometer Count: number of critics rated
  - Audience Score: rating score by audiences
  - Audience Count: number of audiences rated
  - Classification: suitability for certain audiences based on its content
  - Runtime: movie's length
  - Year: release year
  - Original Language: Main language



2.2 Data Preprocessing

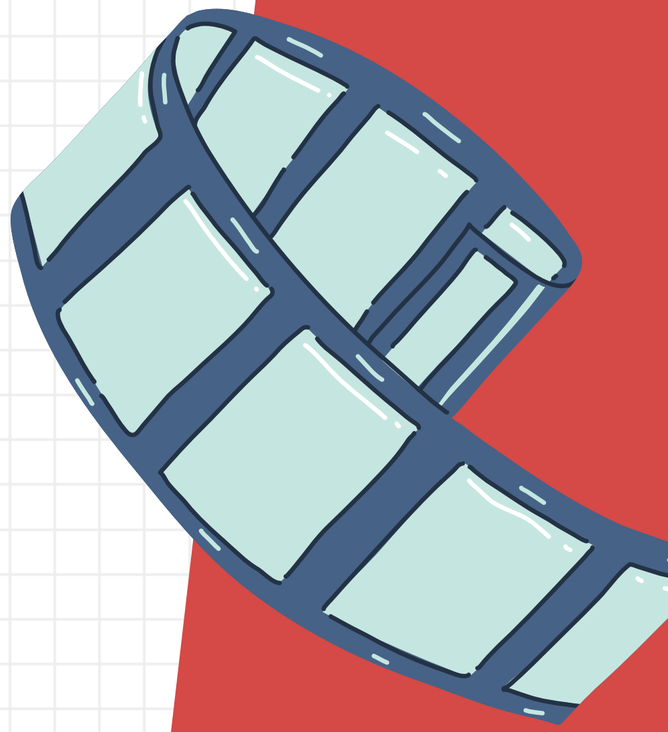


- Check duplicated rows
- Convert data types:
  - release_year (to string)
- Pick first value for genre, classification
- Missing values in classification column

	name	genre	classification	original_language
missing_ratio	0.0	0.0	19.0	0.0
num_values	1207	21	8	27
value_ratios	{'Risen': 0.2, 'Pinocchio': 0.2, 'Halloween': ...	{'Kids & family': 12.8, 'Comedy': 8.8, 'Action...	{'R': 44.4, 'PG-13': 31.5, 'PG': 17.8, 'G': 4....	{'English': 89.0, 'Japanese': 4.0, 'English (U...

-> replace missing values for classification with “Not Rated”

2.2 Data Preprocessing



- Convert Tomatometer_score, tomatometer_count, Audience_score, Audience_count, runtime to **numeric**
- Tomatometer_score, Audience_score to float score:
  - 0% - 100% = 0.0 - 1.0
- Convert runtime to minutes

3. Data Exploration

Question 1: Does the runtime of a movie or TV series depend on its classification?

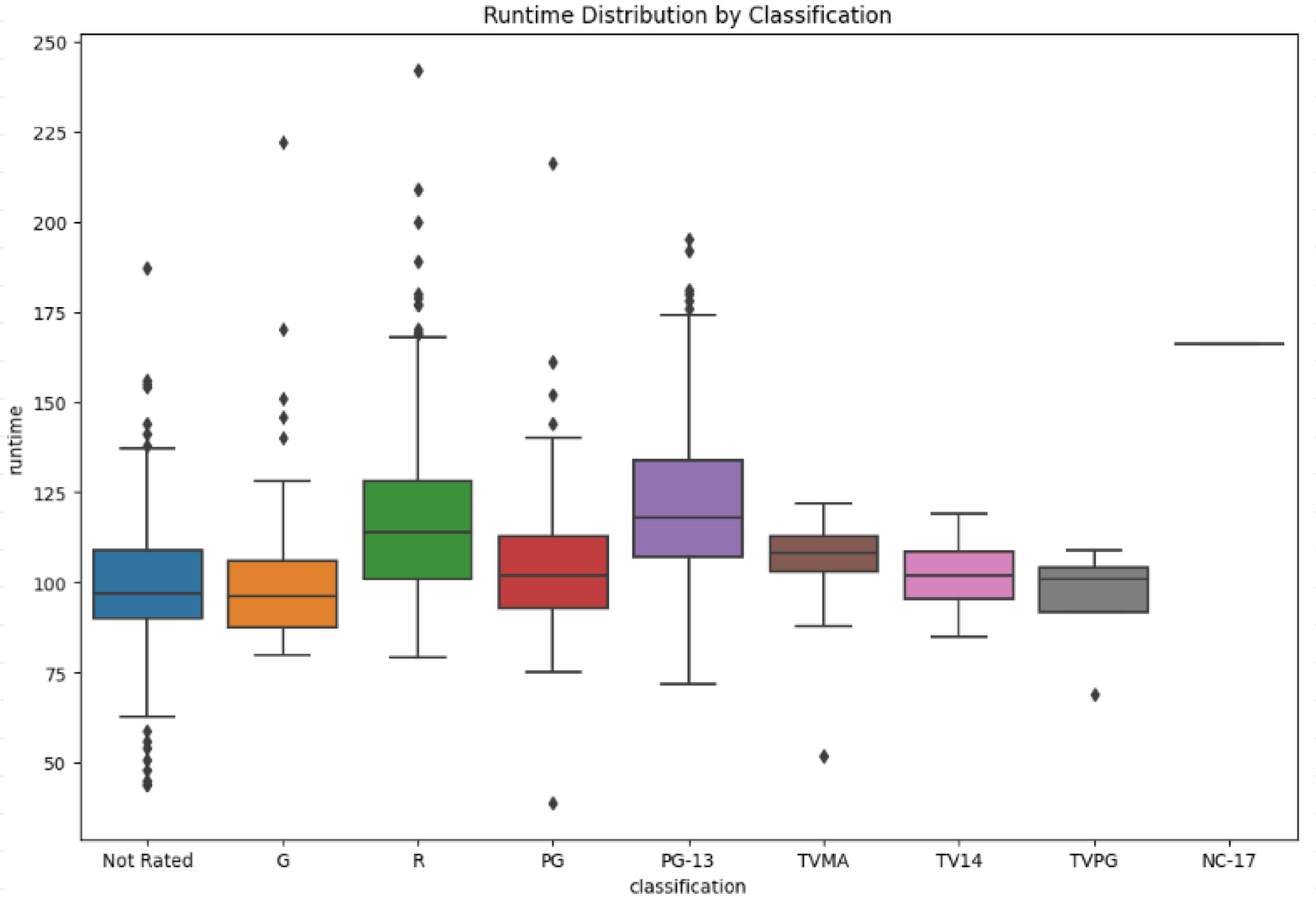
Question 2: Are there any correlations between age of a movie and its ratings?

Question 3: For each genre, what would the correlation between tomatometer score and audience be ?

Question 4: What genre would people/ critics prefer to watch (audience/ tomatometer count) ?

Question 5: What is the highest rate for a genre and a classification to be in a movie together?

Does the runtime of a movie or TV series depend on its classification?



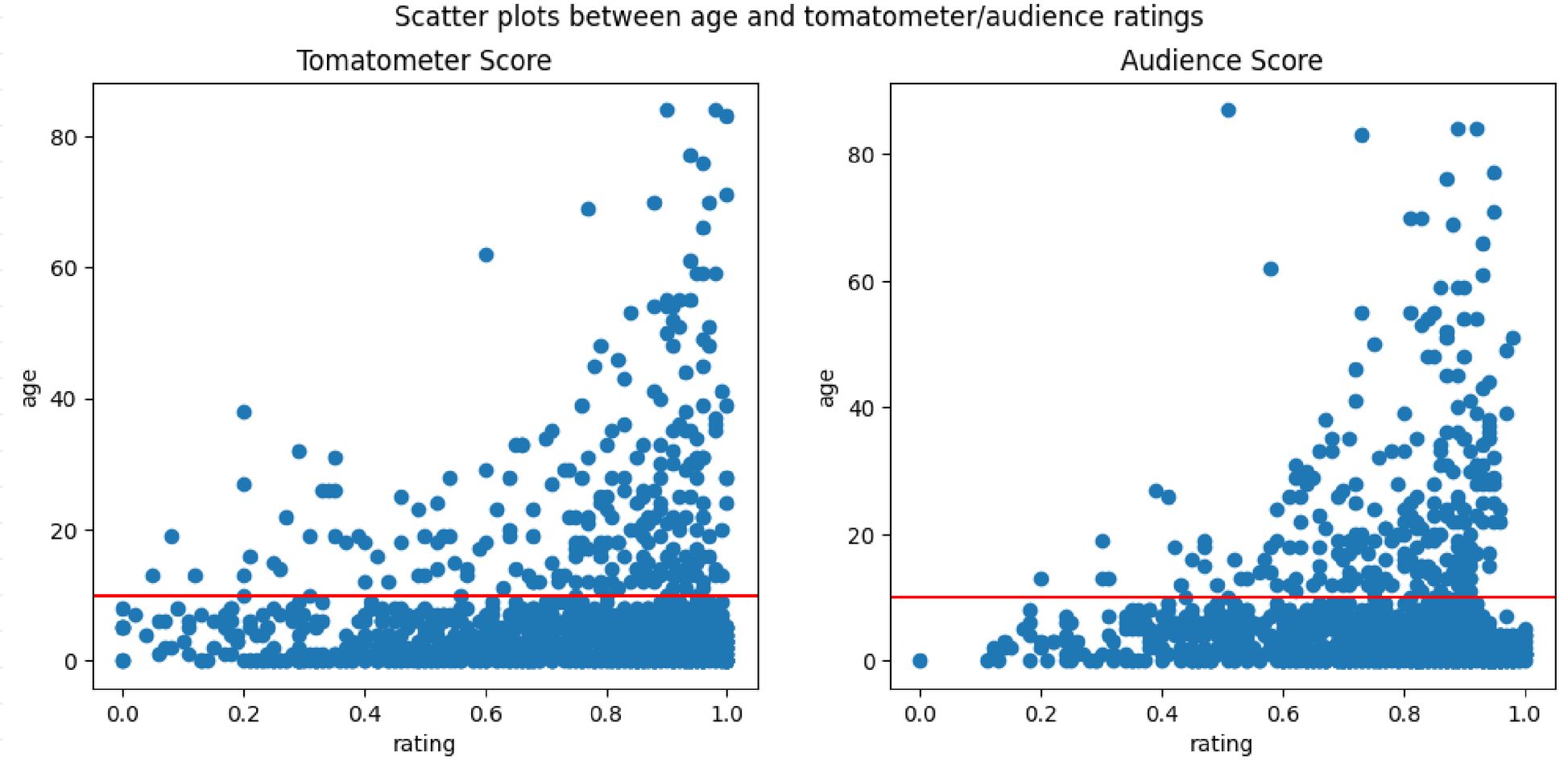
Does the runtime of a movie or TV series depend on its classification?

Applying **null hypothesis** :

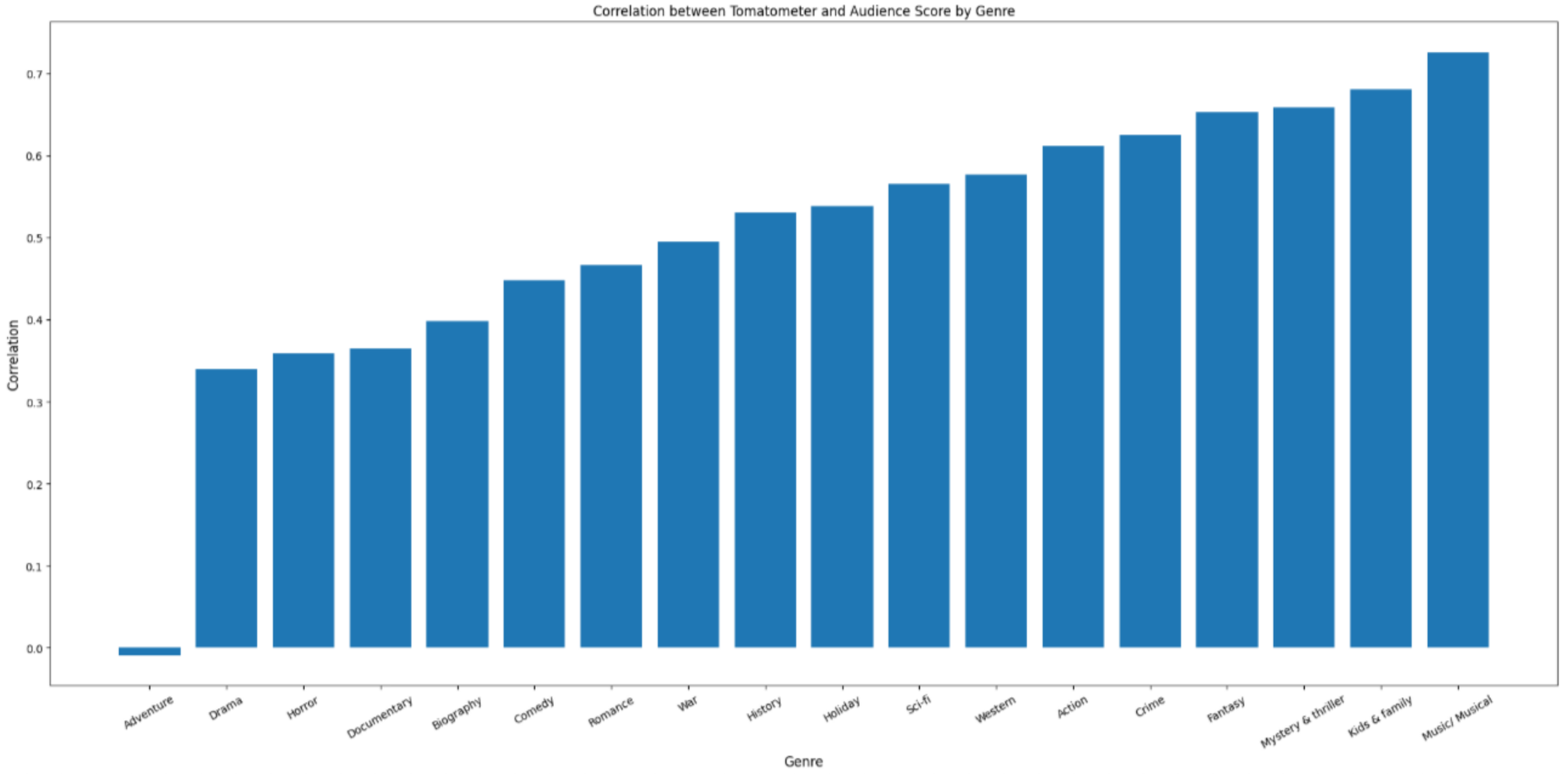
- H0: u1=u2=u3=...=uk Where:
  - H0 is the null
  - u1,u2,...,uk:mean of the runtime for different classifications.
- This null hypothesis assumes that there is no significant difference in runtime among different classifications.

-> ANOVA p-value: 1.2597456890142491e-17

Are there any correlations between age of a movie and its ratings?



For each genre, what would the correlation between tomatometer score and audience be ?



For each genre, what would the correlation between tomatometer score and audience be ?

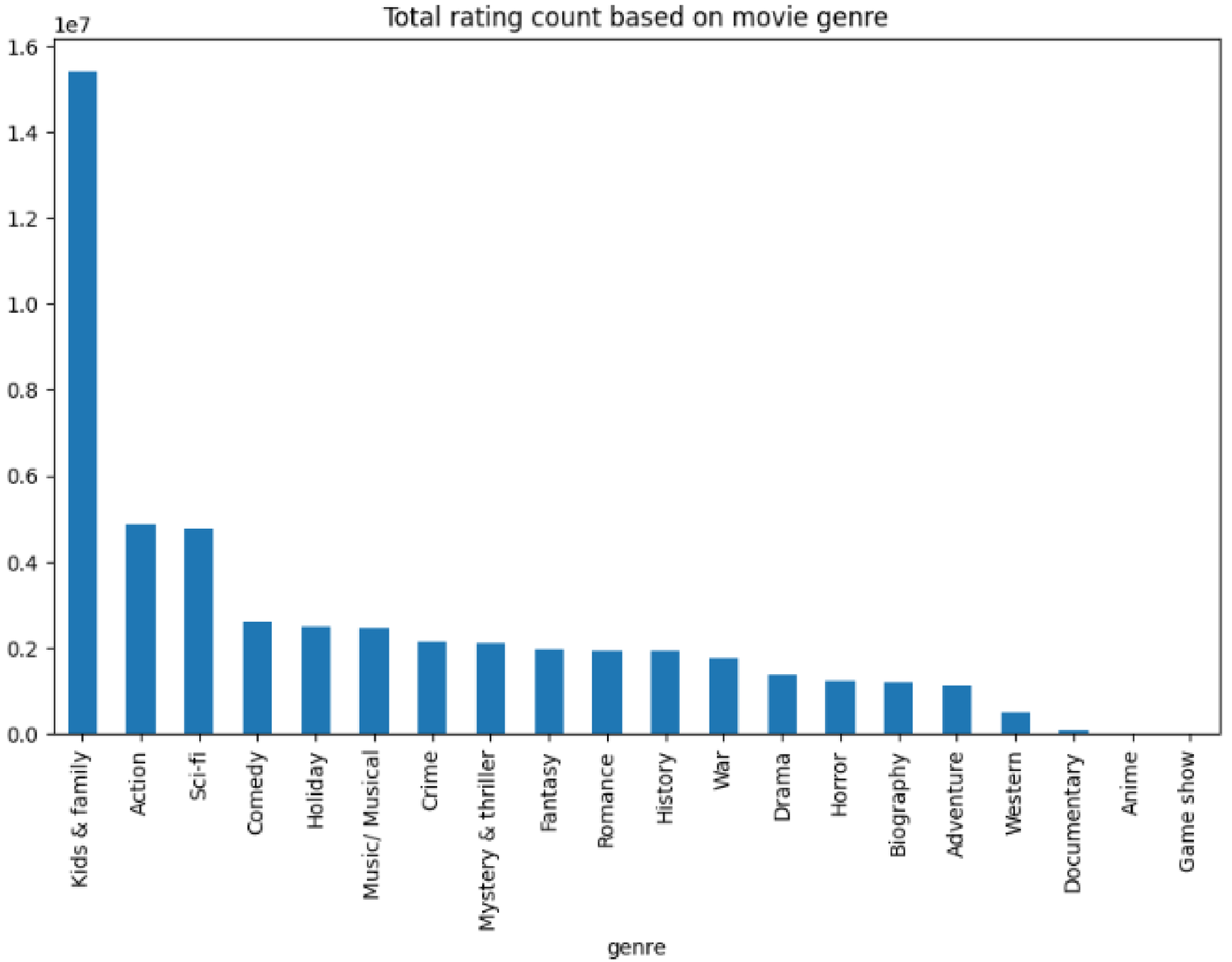
Insight:

- Genre with highest correlation: Music/Musical
- Genre with lowest correlation: Adventure

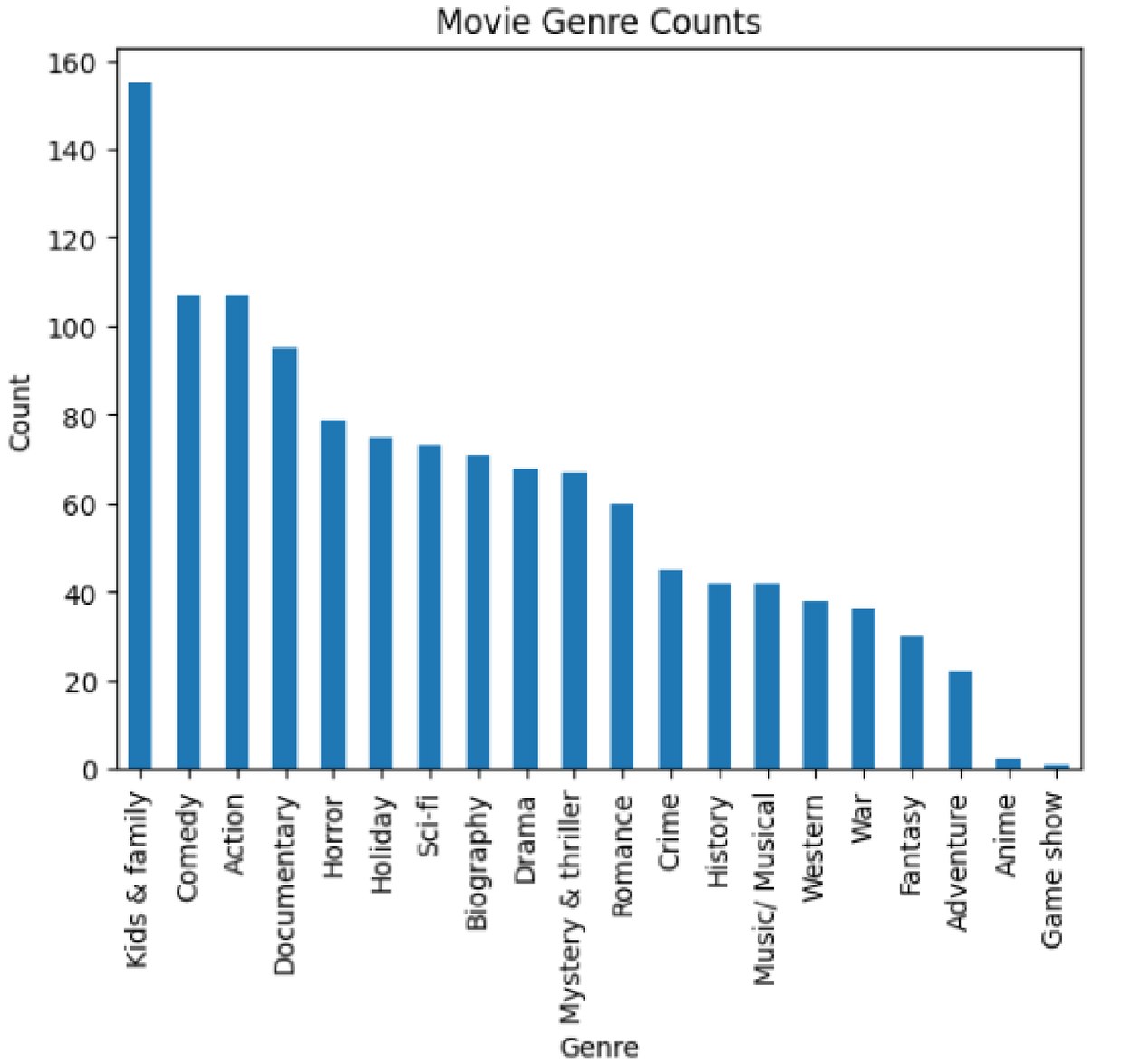
Why Music/Musical has the highest correlation?

Why Adventure has the lowest correlation?

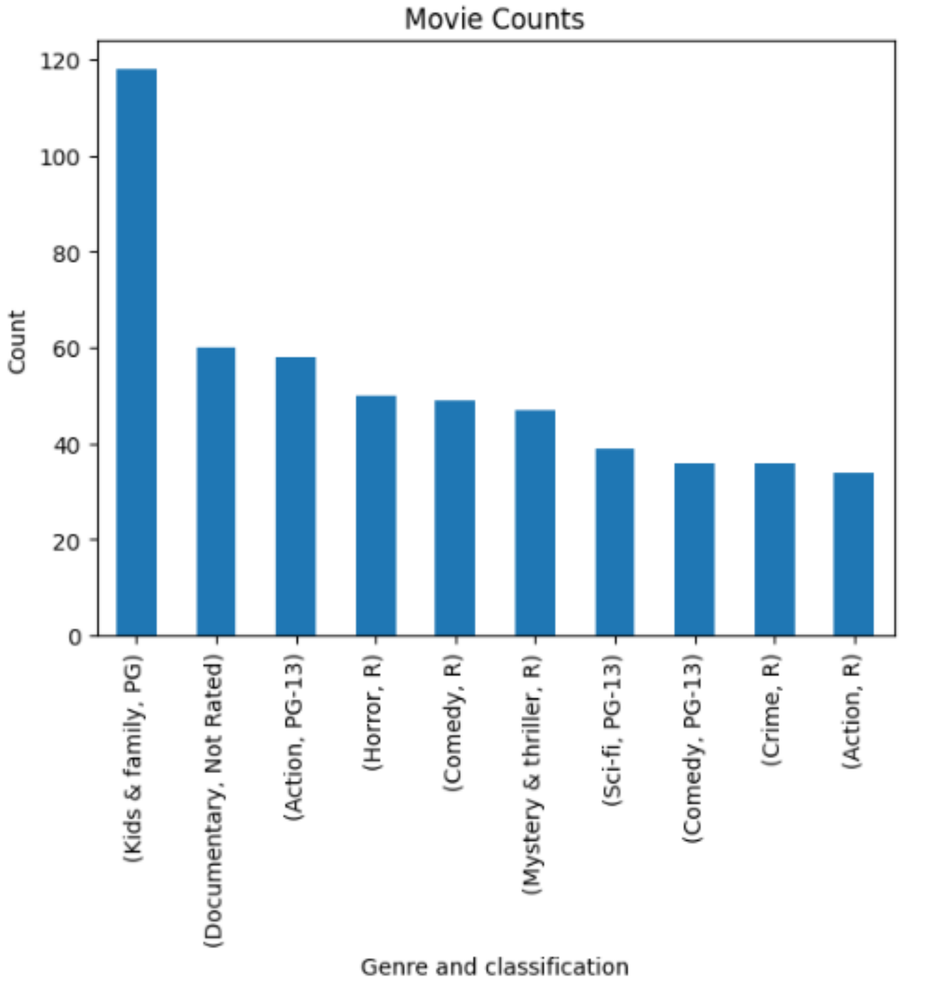
What genre would people/ critics prefer to watch?



What genre would people/ critics prefer to watch?



What is the highest rate for a genre and a classification to be in a movie together?



genre	classification	
Kids & family	PG	9.711934
Documentary	Not Rated	4.938272
Action	PG-13	4.773663
Horror	R	4.115226
Comedy	R	4.032922
		...
	TVMA	0.082305
Horror	TVMA	0.082305
Music/ Musical	TV14	0.082305
Comedy	TV14	0.082305
Western	TVMA	0.082305

Length: 90, dtype: float64

4. Data Modeling

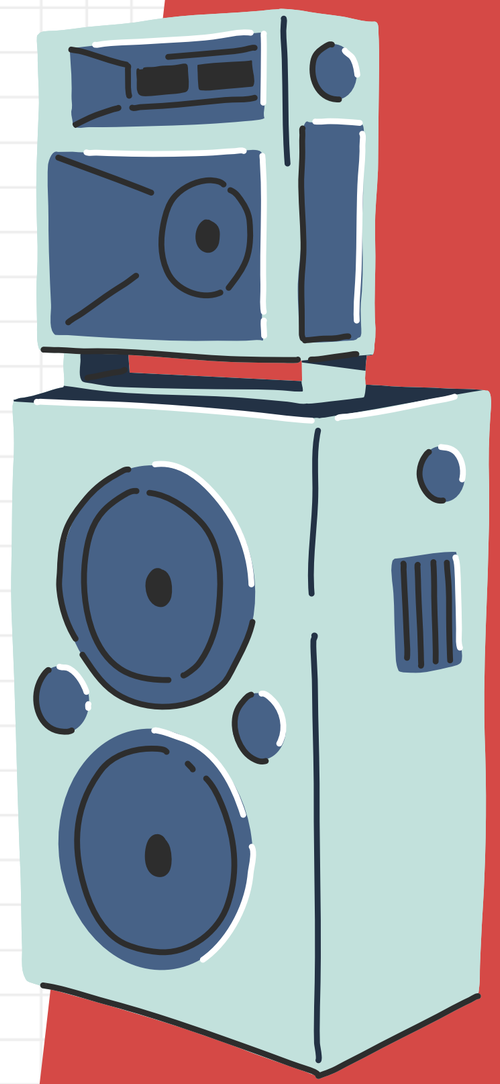
Problem: Predict audience ratings

- This problem can be useful in building recommender systems, trend forecasting,...

Metrics: Mean Squared Error

We choose to split the data for 3 parts:

- Training: 70%
- Validation: 20%
- Testing: 10%



4.1. Linear Regression

Model's advantages:

- Simplicity and interpretability
- Efficiency: can be trained quickly

Running process:

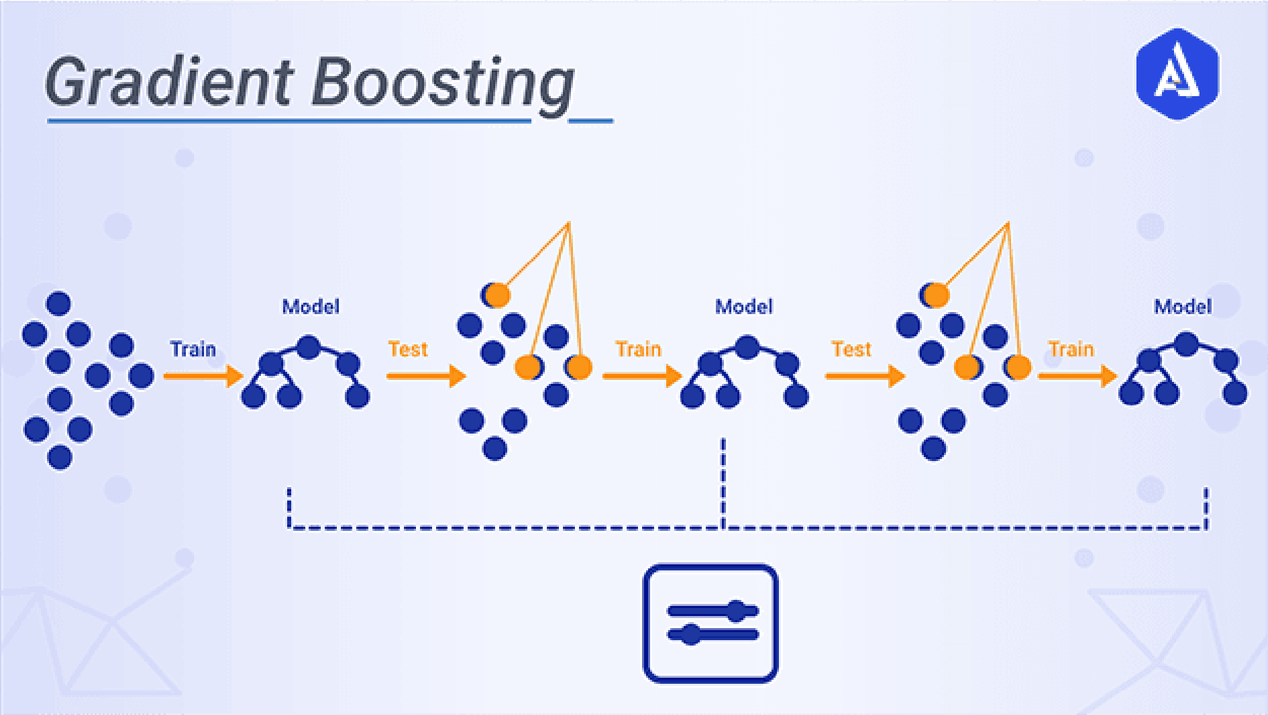
MSE: 0.02156328525264906

4.2 Gradient Boosting Regressor

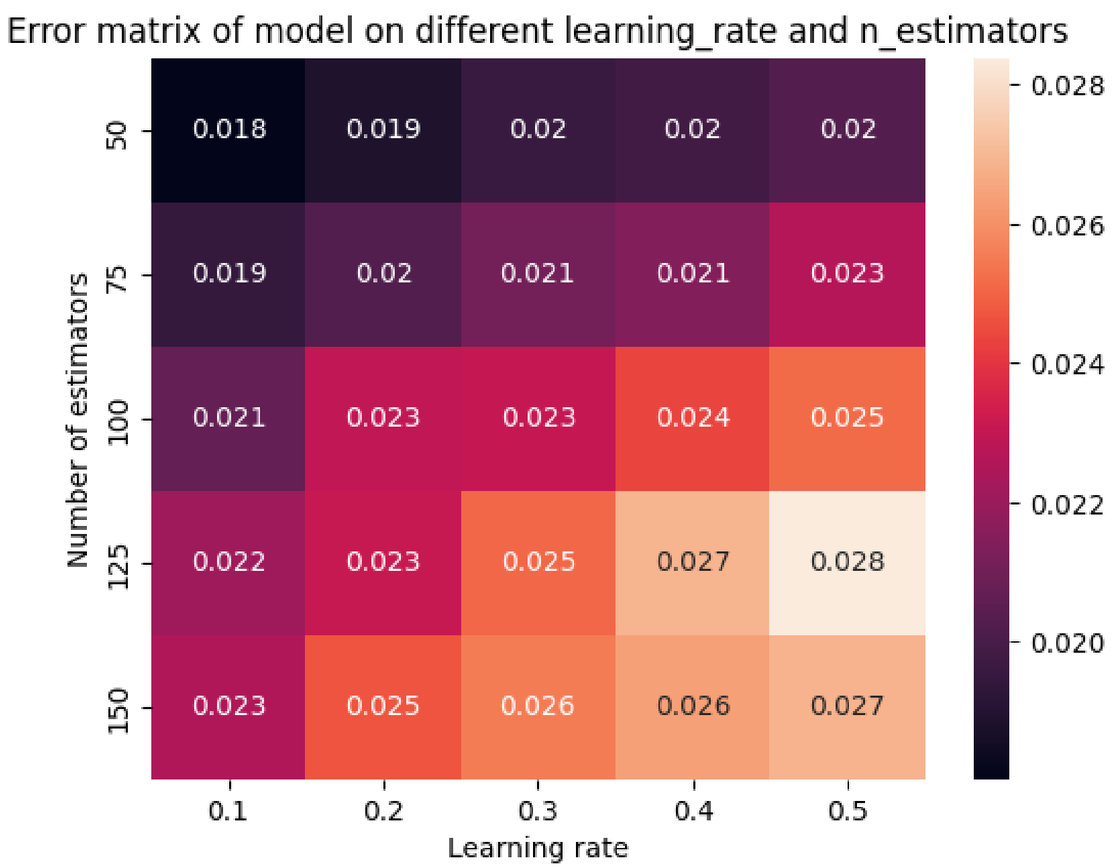
GradientBoostingRegressor from *sklearn*

Model's advantages:

- Flexible to outliers
- Works well for small sized data



4.2 Gradient Boosting Regressor

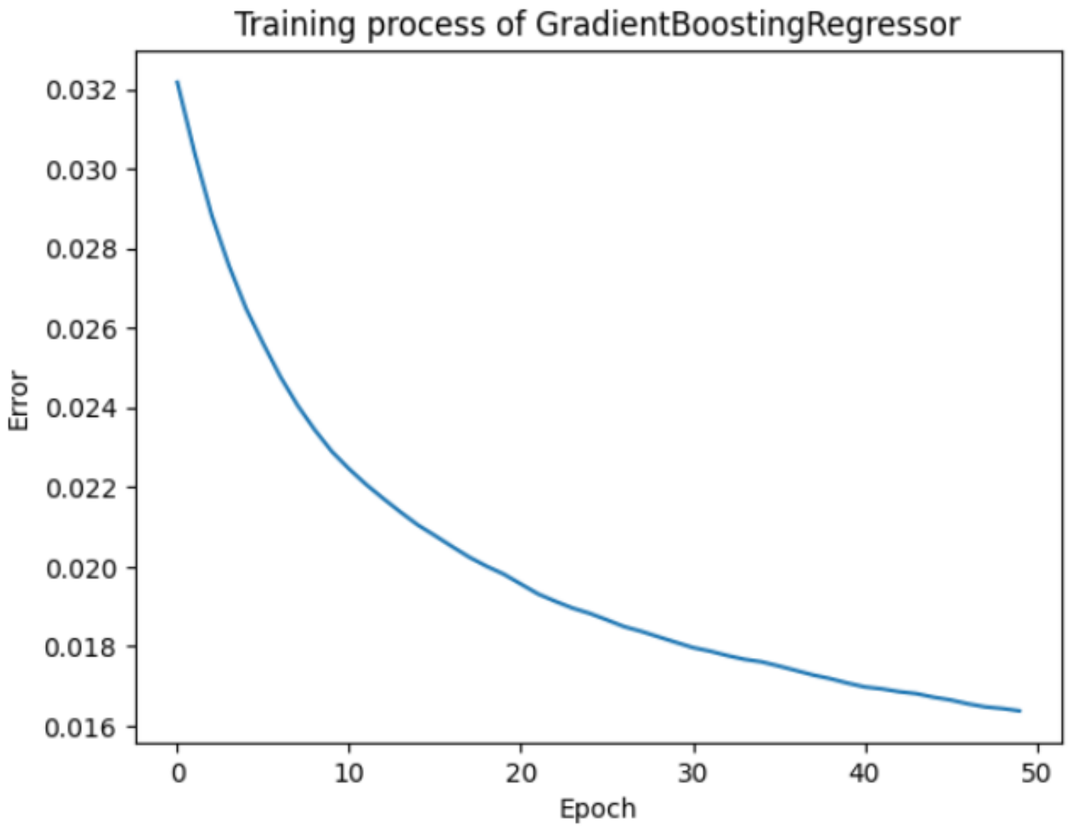


Fine-tuning:

*P*erform hyperparameter tuning: learning_rate and n_estimator.

So, the best hyperparameters are learning_rate = 0.1 and n_estimators = 50.

4.2 Gradient Boosting Regressor



Running process:

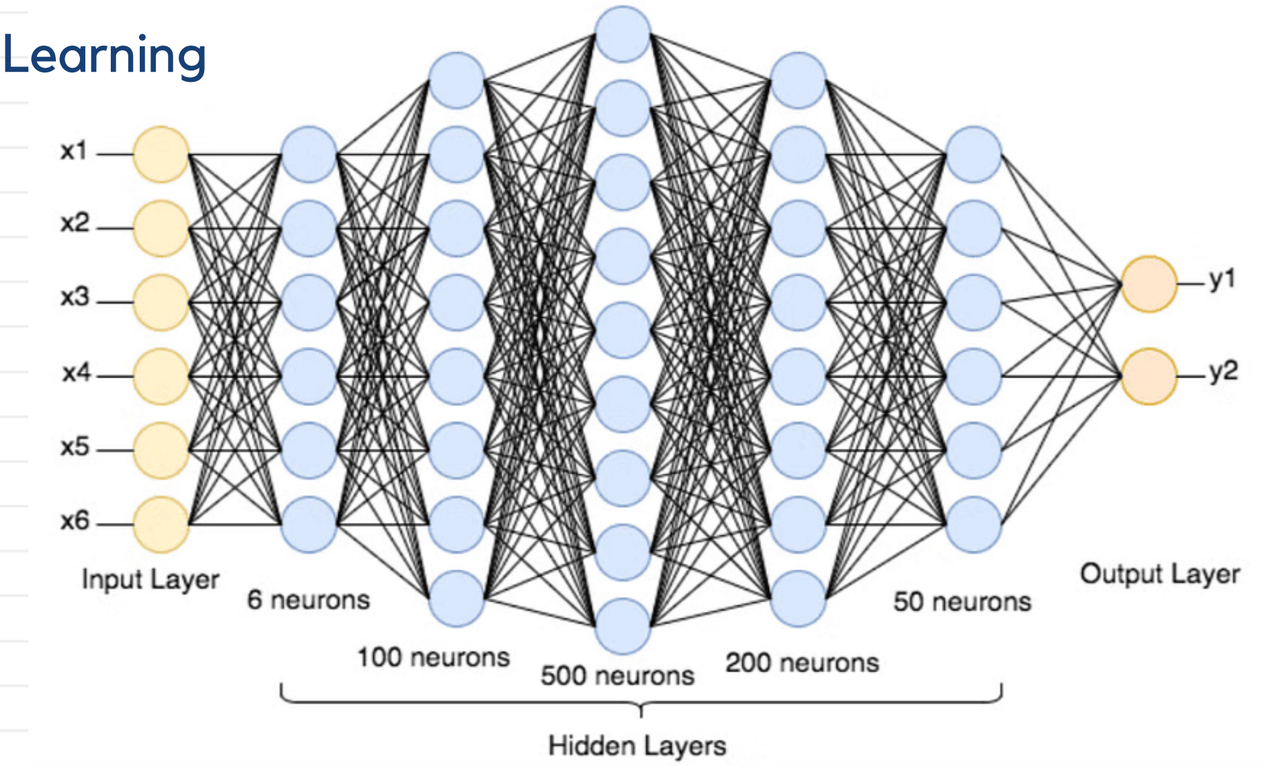
MSE: 0.0201136720028379 3

4.3 Neural Network

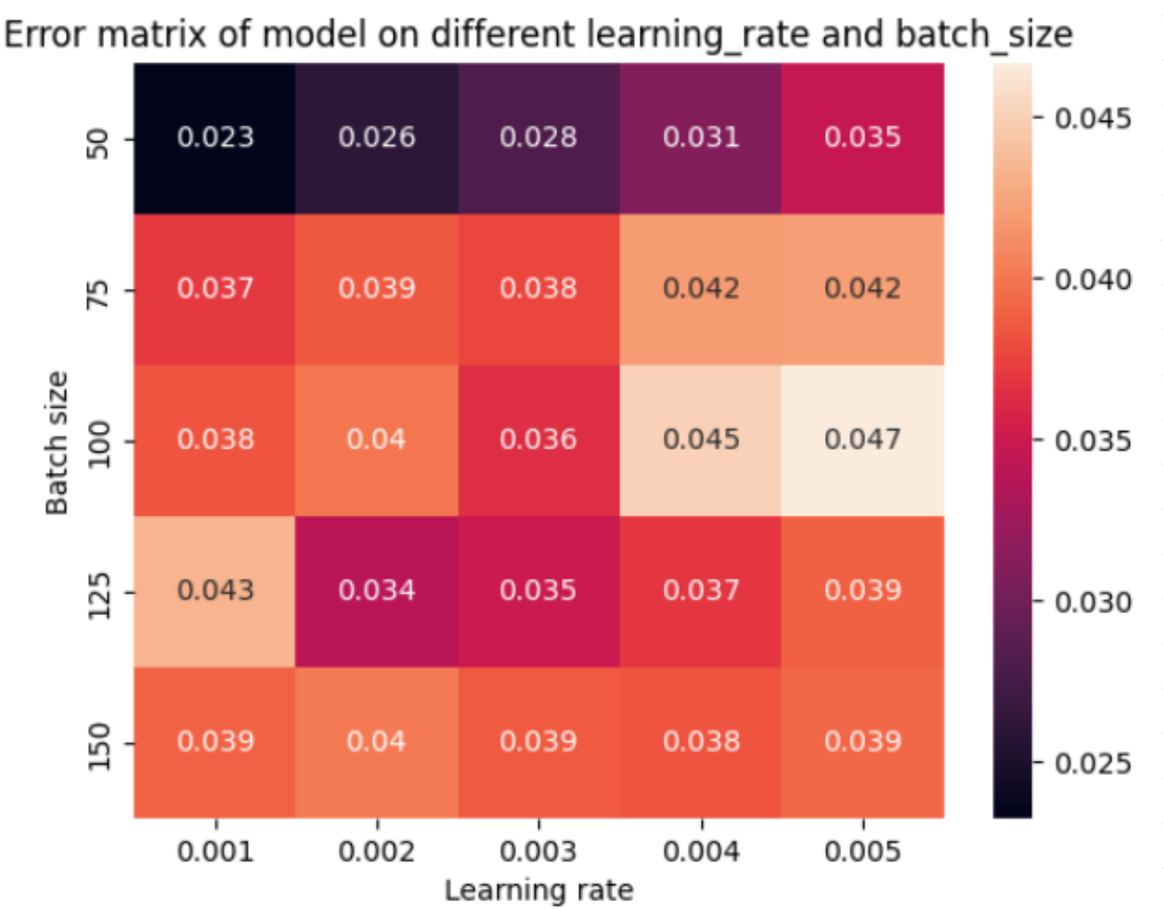
NeuralNetwork from *tensorflow*

Advantages of this model

- Feature Interactions:
- Representation Learning



4.3 Neural Network

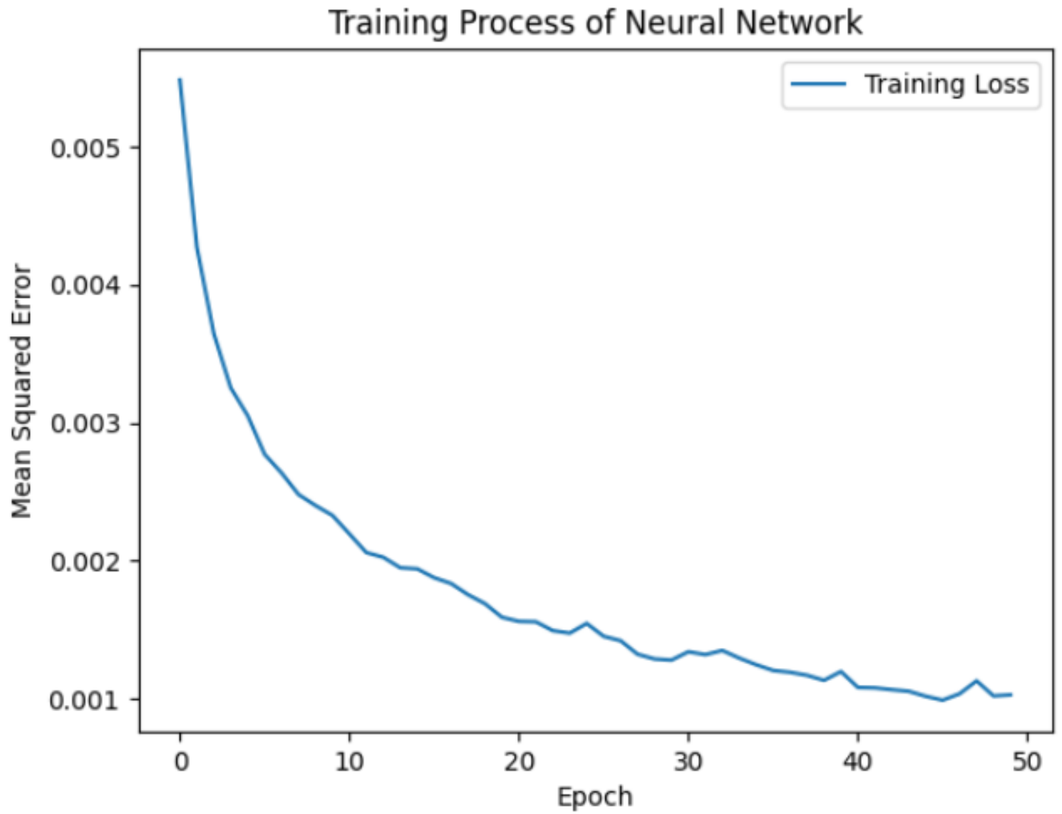


Fine-tuning:

*P*erform hyperparameter tuning: learning_rate and batch_size

So, the best hyperparameters are learning_rate = 0.001 and batch_size = 50.

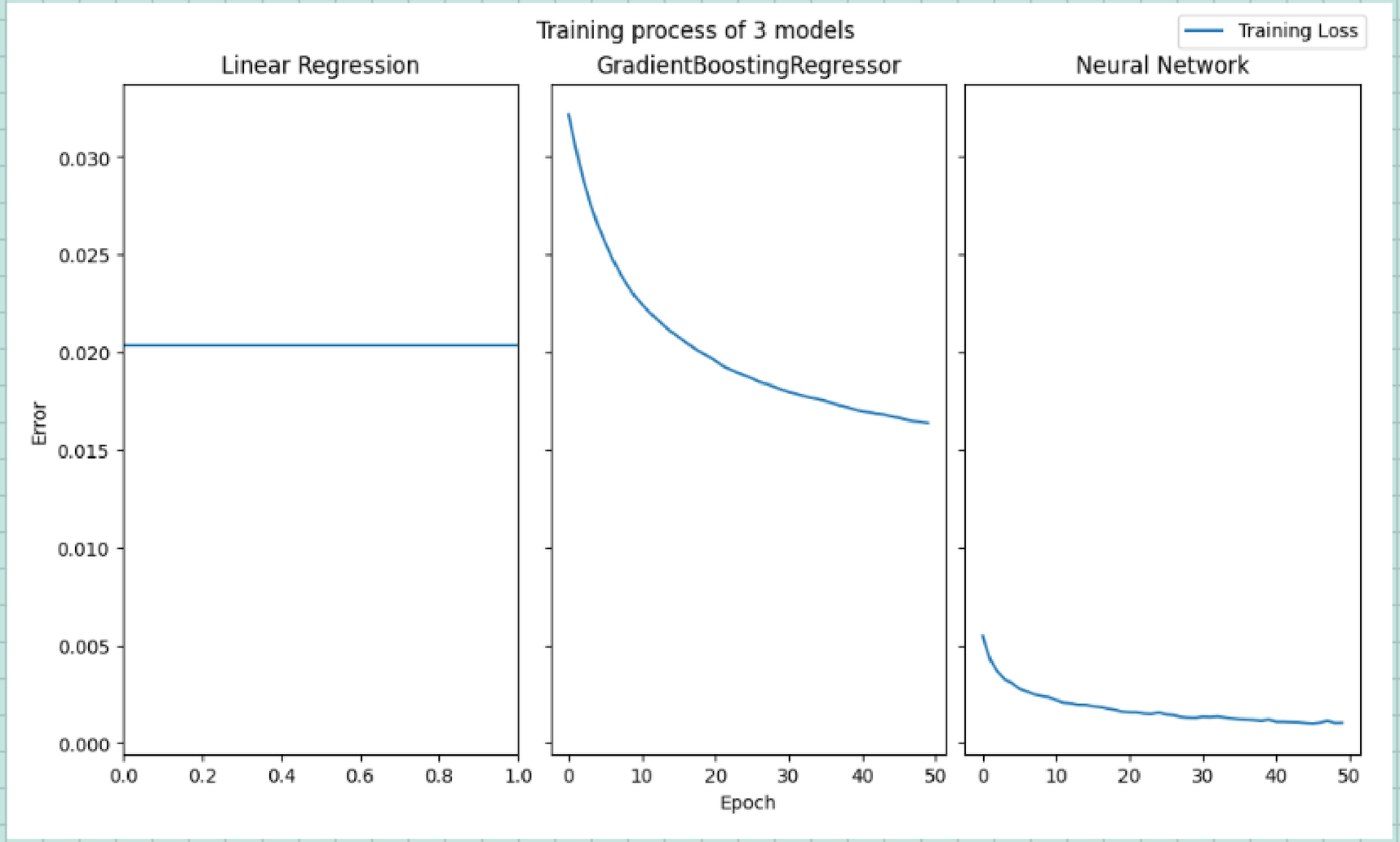
4.3 Neural Network



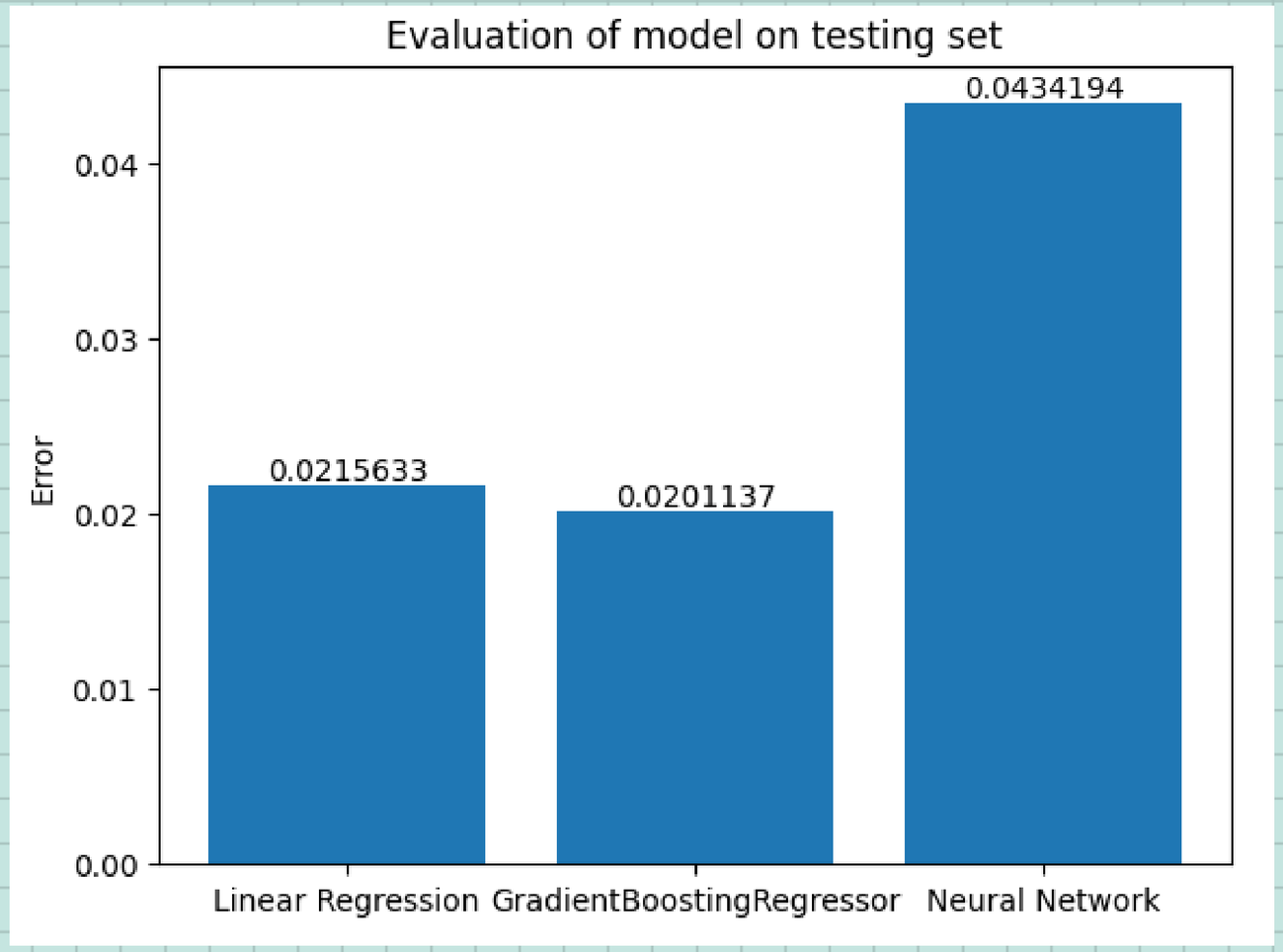
Running process:

MSE: 0.0434194

Comparing models

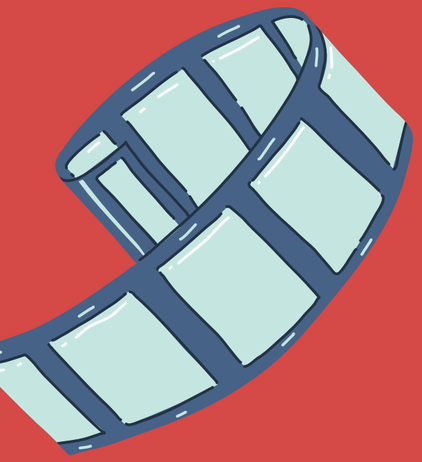


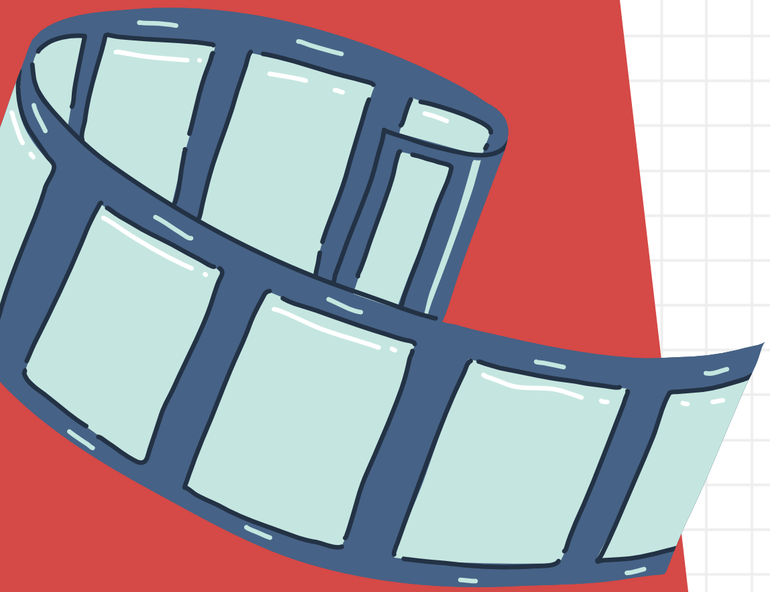
Comparing models

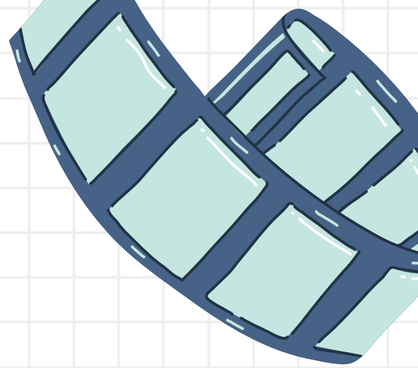


- Training:
  - Linear Regression: fastest training time.
  - Neural Network takes a longer time through each epoch, but seems to learn faster than GradientBoostingRegressor.
- Testing Evaluation:
  - Neural Network: highest testing error.

  -> NN model has been overfitted to training data.
  - Linear Regression & GradientBoostingRegressor have low testing error.







Data modeling summary

- For a small dataset, Neural Network is very inefficient and becomes very likely to overfit/underfit

  -> apply 'early-stopping' methods to minimize this.
- Linear Regression works well, but might lose accuracy when the dataset becomes larger and more complex.
- GradientBoostingRegressor is also sufficient, but it takes more epochs in order to reach the best possible state, which might be time-consuming as the dataset grows.

Thank you for listening