What makes a Good Anime?

Team 7
Caleb Yeo (U2323500G)
Ong Rie Ian (U2320887J)
Toh Xin Yi (U2321731B)

Problem Definition

What makes a Good Anime?

What elements in Anime constitutes a high audience approval?

Can we successfully predict the next grossing Anime?

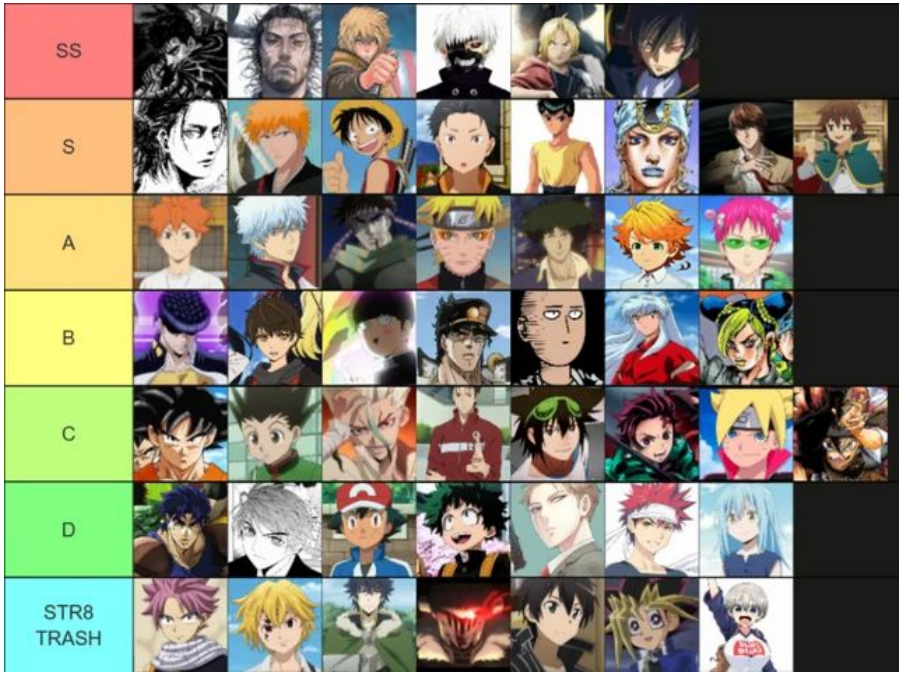
Problem Definition

We decided to quantify the quality of the Anime via its ratings

Animes with high ratings meant that it received a positive review from a large majority of critics

These Animes align well with certain preferences of the general public, which perhaps can indicate how ‘good’ it is

Motivation



As *Anime enthusiasts*, we've seen reviews of the top Anime hits of the year online

We are curious to the reason behind the success of these Anime — are there any common elements between them? Does having a specific element improve its audience's perception?

Through this, we also hope to find the *recipe* for the next Top Anime

Dataset: Anime.csv (Kaggle)

Anime Recommendations Database

Recommendation data from 76,000 users at myanimelist.net

Data Card | Code (308) | Discussion (25) | Suggestions (0)

About Dataset

Context

This data set contains information on user preference data from 73,516 users on 12,294 anime. Each user is able to add anime to their completed list and give it a rating and this data set is a compilation of those ratings.

Usability ⓘ
8.24

License
CC0: Public Domain

Expected update frequency
Not specified

Dataset: Anime.csv (Kaggle)

	anime_id	name	genre	type	episodes	rating	members
0	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9.37	200630
1	5114	Fullmetal Alchemist: Brotherhood	Action, Adventure, Drama, Fantasy, Magic, Mili...	TV	64	9.26	793665
2	28977	Gintama°	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9.25	114262
3	9253	Steins;Gate	Sci-Fi, Thriller	TV	24	9.17	673572
4	9969	Gintama'	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9.16	151266
...	...	...	...	...	...	...	...
12289	9316	Toushindai My Lover: Minami tai Mecha-Minami	Hentai	OVA	1	4.15	211
12290	5543	Under World	Hentai	OVA	1	4.28	183
12291	5621	Violence Gekiga David no Hoshi	Hentai	OVA	4	4.88	219
12292	6133	Violence Gekiga Shin David no Hoshi: Inma Dens...	Hentai	OVA	1	4.98	175
12293	26081	Yasuji no Pornorama: Yacchimae!!	Hentai	Movie	1	5.46	142

12294 rows × 7 columns

Data Cleaning: Replacing ‘Unknown’ data

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12294 entries, 0 to 12293
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   anime_id  12294 non-null  int64
 1   name      12294 non-null  object
 2   genre     12232 non-null  object
 3   type      12269 non-null  object
 4   episodes  12294 non-null  object
 5   rating    12064 non-null  float64
 6   members   12294 non-null  int64
dtypes: float64(1), int64(2), object(4)
memory usage: 672.5+ KB
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12294 entries, 0 to 12293
Data columns (total 6 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   anime_id  12294 non-null  int64
 1   name      12294 non-null  object
 2   genre     12232 non-null  object
 3   type      12269 non-null  object
 4   episodes  12294 non-null  int64
 5   rating    12064 non-null  float64
dtypes: float64(1), int64(2), object(3)
memory usage: 576.4+ KB
```

- Removed ‘Unknown’ values in ‘episode’ column
  - ‘Unknown’ == 340
- Calculated the mean episode count of known values
- Replaced ‘Unknown’ with mean values
- Convert ‘episode’ column back into integer type

Data Cleaning: Combining ‘Type’

```
#combined types into others for convenience
anime['type'].replace(['ONA', 'Music', 'Special'], 'Others', inplace=True)
anime.dropna(subset=['type'], inplace=True) #removed 25 NaN values
print(anime['type'].value_counts())
print(anime['type'].unique())
anime.info()
```

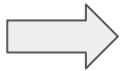
```
type
TV        3787
OVA       3311
Others    2823
Movie     2348
Name: count, dtype: int64
['Movie' 'TV' 'OVA' 'Others']
```

- Combined anime types ‘ONA’, ‘Music’, and ‘Special’ into a general type named ‘Others’

Data Cleaning: Dropping Genre Values

```
anime = anime[~anime['genre'].str.contains('Hentai|Yuri|Yaoi|Shounen Ai|Shoujo Ai', na=False)]
```

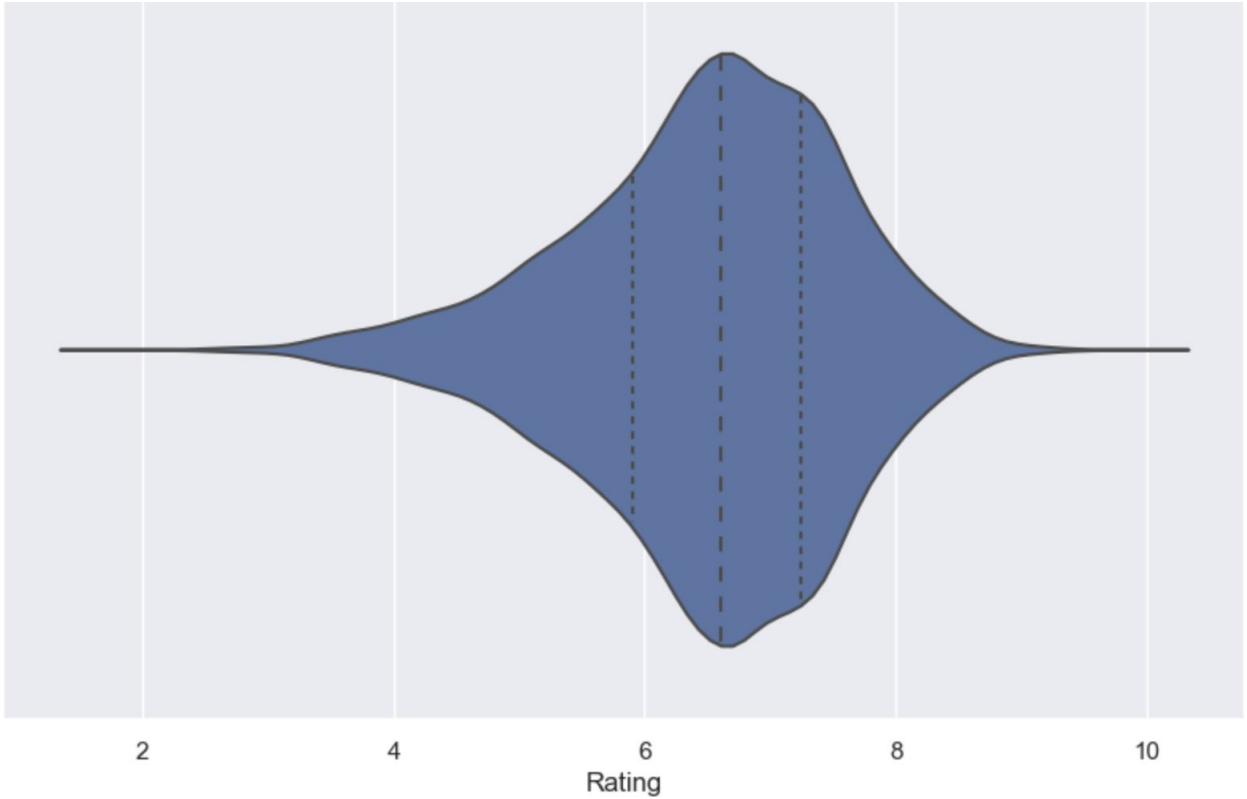
```
<class 'pandas.core.frame.DataFrame'>
Index: 12269 entries, 0 to 12293
Data columns (total 6 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   anime_id  12269 non-null  int64
 1   name      12269 non-null  object
 2   genre     12210 non-null  object
 3   type      12269 non-null  object
 4   episodes  12269 non-null  int64
 5   rating    12064 non-null  float64
dtypes: float64(1), int64(2), object(3)
memory usage: 671.0+ KB
```



```
<class 'pandas.core.frame.DataFrame'>
Index: 10914 entries, 0 to 11111
Data columns (total 6 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   anime_id  10914 non-null  int64
 1   name      10914 non-null  object
 2   genre     10914 non-null  object
 3   type      10914 non-null  object
 4   episodes  10914 non-null  int64
 5   rating    10733 non-null  float64
dtypes: float64(1), int64(2), object(3)
memory usage: 596.9+ KB
```

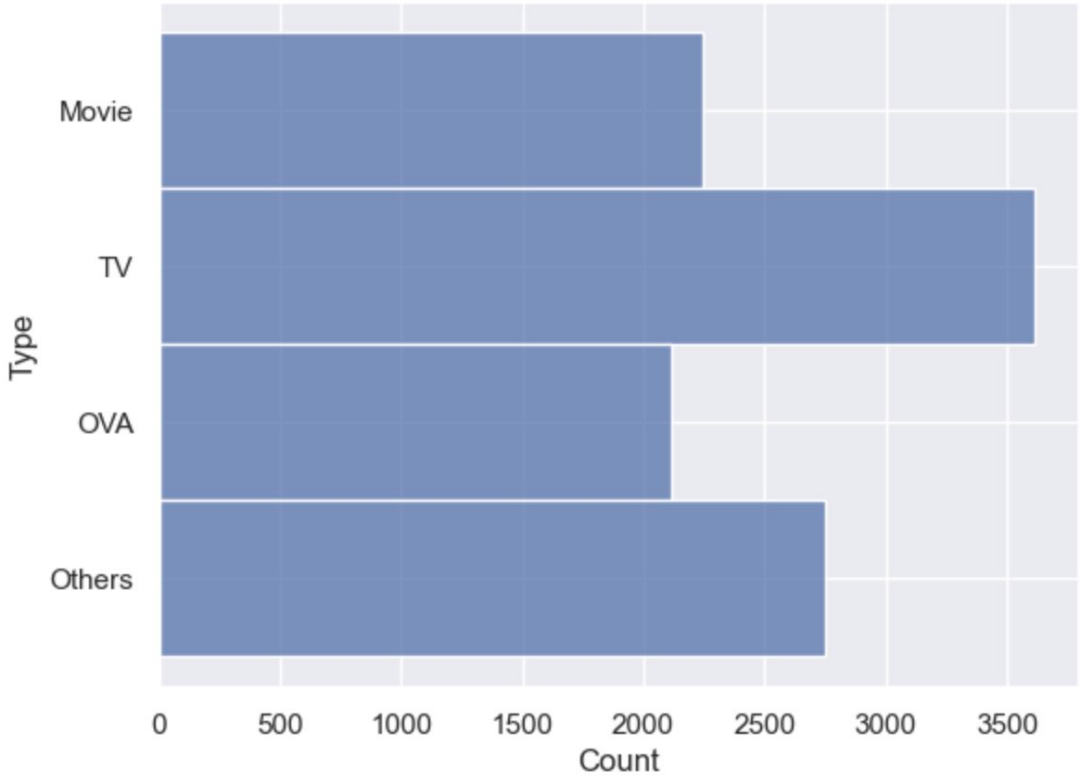
- Dropped certain inappropriate genre values from the dataset

Exploratory Data Analysis: Violin Plot of Anime Ratings



- Variance of ratings: 1.10 (3 s.f.)
- Majority of the Animes have a rating of 6-7

Exploratory Data Analysis: Bar Chart of Anime Types

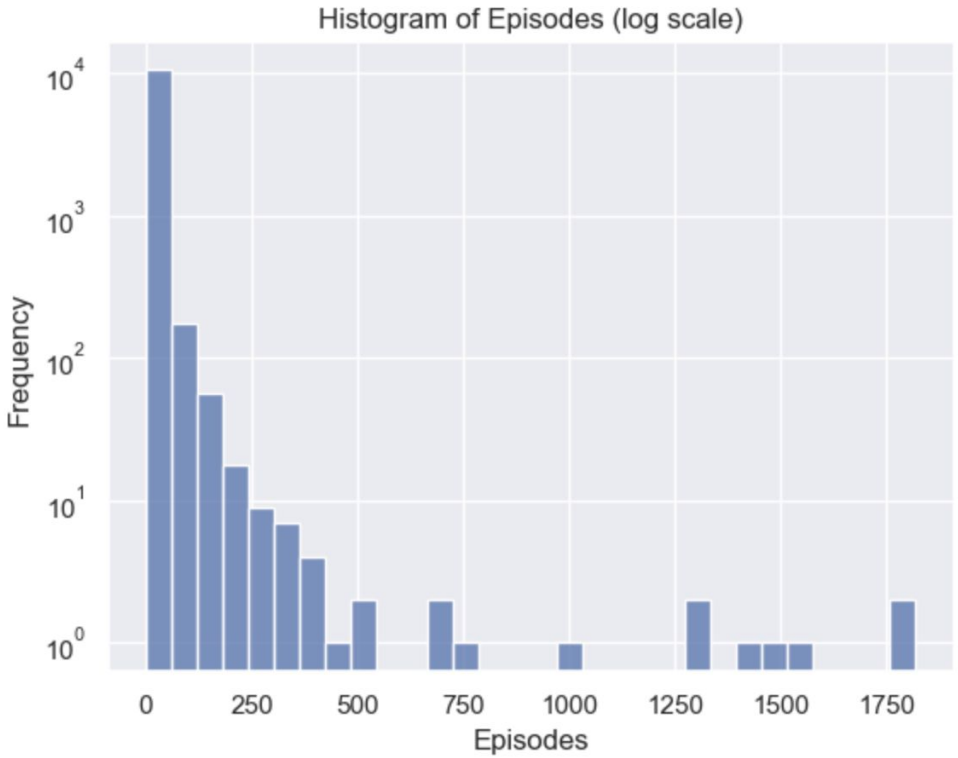


- Successful replacement of certain data types to ‘Others’
- Dataset has the highest count of Animes of type ‘TV’

Exploratory Data Analysis: Violin Plot of Type vs Rating

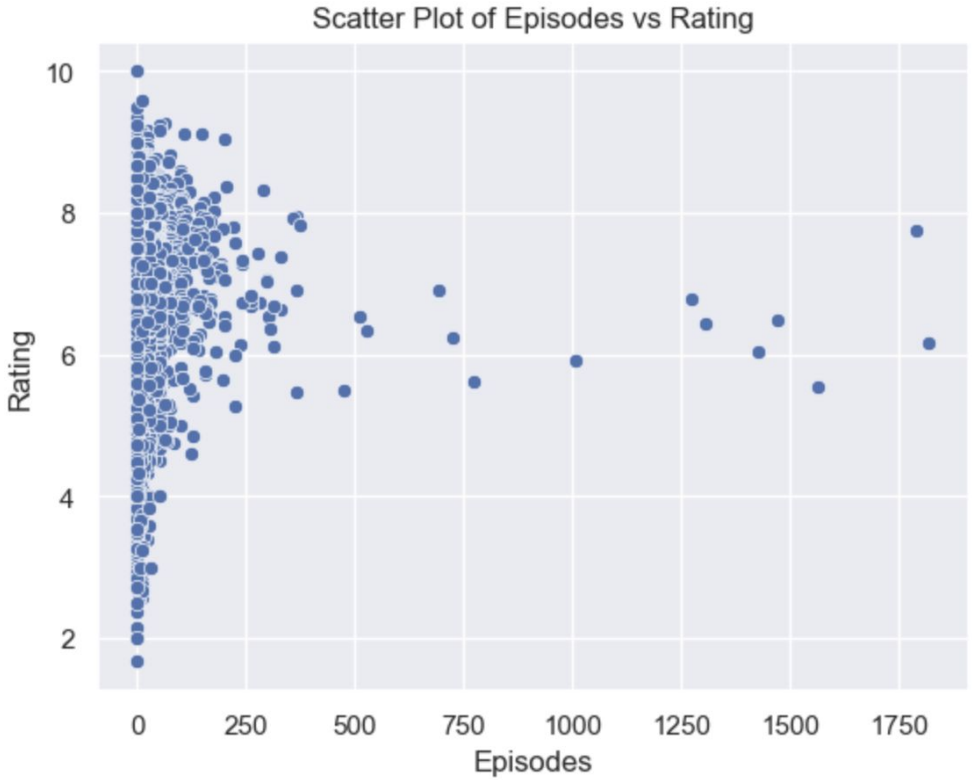
- We noticed the different spread and medians of the different types
- ‘Movie’ and ‘TV’ generally have a higher rating, with a higher maximum rating

Exploratory Data Analysis: Histogram of Episode Count



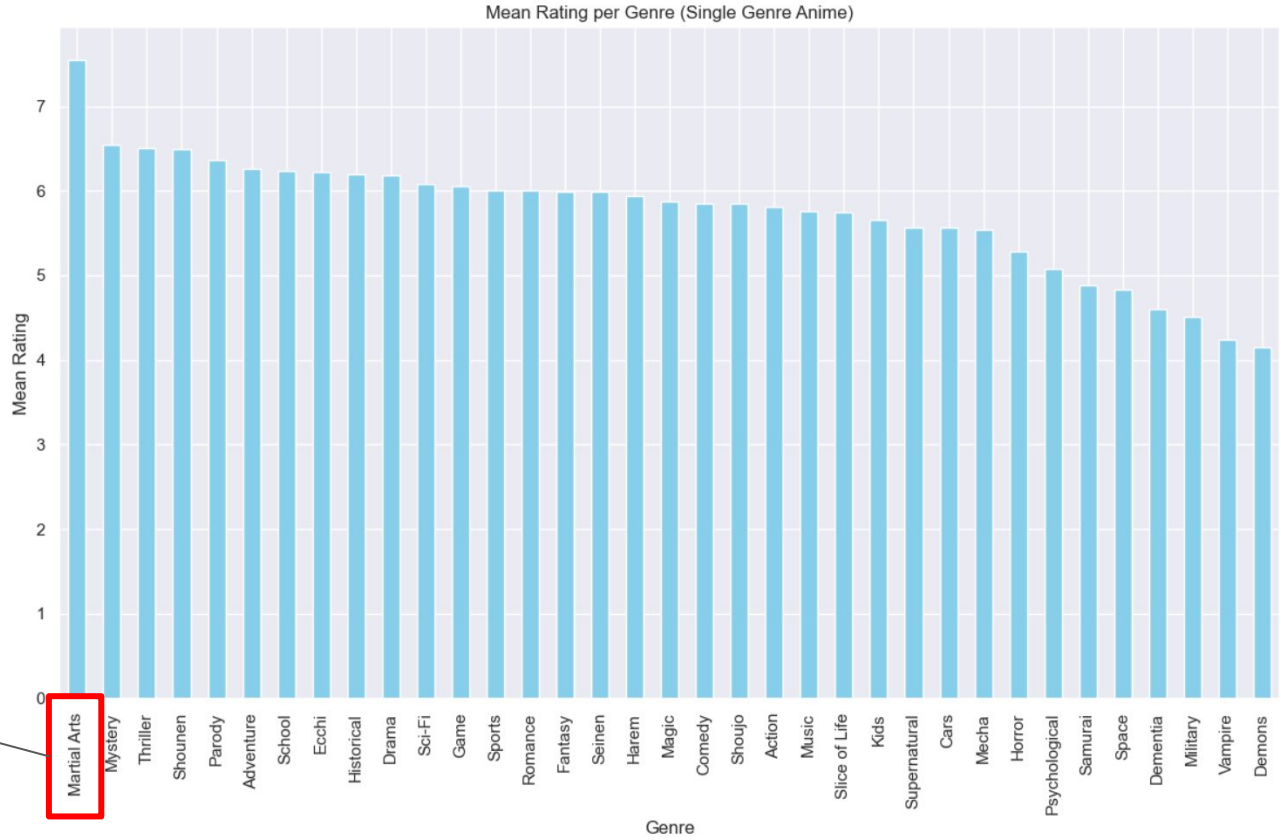
- We applied a logarithmic scale so we can fit the graph more accurately

Exploratory Data Analysis: Scatterplot of Episodes vs Rating



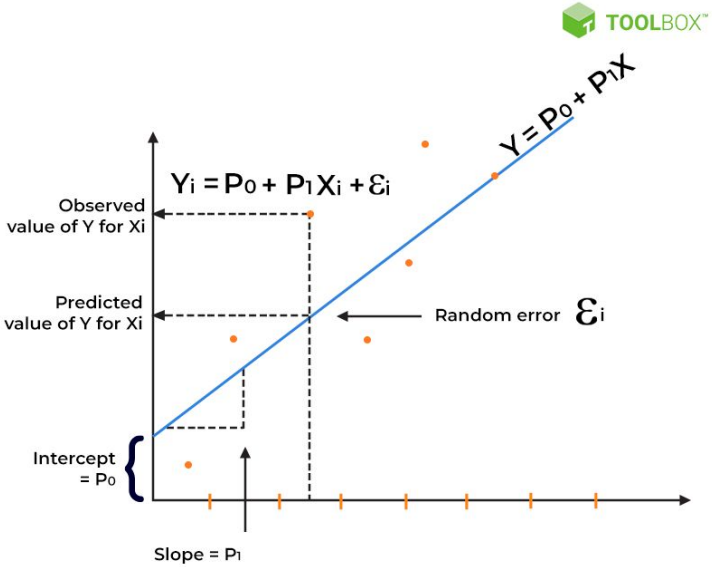
- To determine the relationship between episode count and rating of an Anime
- As seen, most Animes have short episode count, congregating at the left end, with most having a rating of 6-8
- Animes with high episode count have ratings of ~5-7

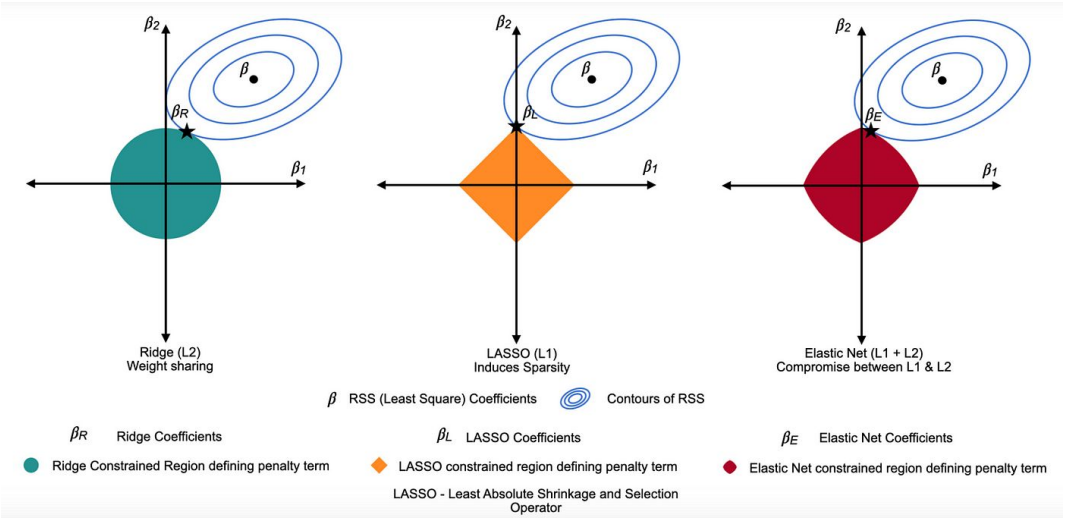
Exploratory Data Analysis: Mean Rating by Genre



Martial Arts

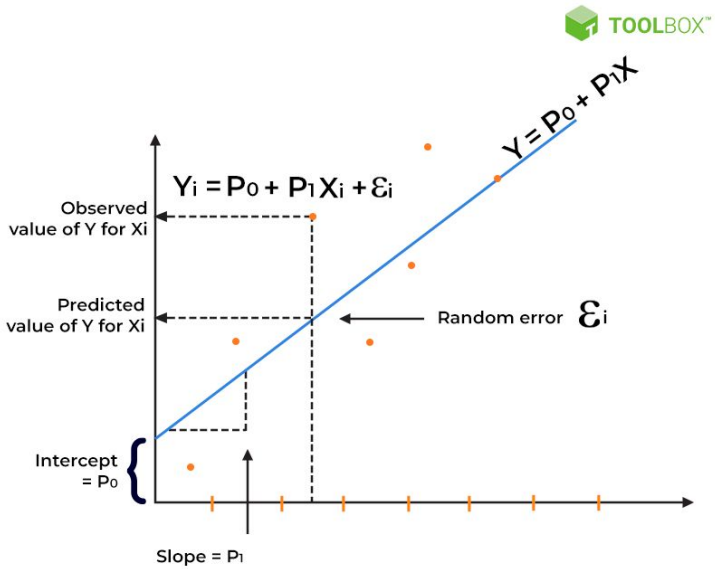
Machine Learning: Summary





- Linear regression (average rating)
- Lasso, ridge, elastic net regression (one-hot encoding)

Machine Learning: Linear Regression Summary



- Linear regression to predict the rating (D.V.) of Animes according to genre, type, and number of episodes (I.V.) an Anime has
- A higher predicted rating signifies a higher positive correlation between I.V. and rating
- This gives us insight on how a certain element (i.e. genre) can make an Anime more likeable

Data Preparation: Filtering Genres

```python
# Drop rows with missing genre values
anime.dropna(subset=['genre'], inplace=True)

# Filter out rows with only one genre
single_genre_index = ~anime['genre'].str.contains(',', na=False)
single_genre_anime = anime.loc[single_genre_index]

# Split the single genre from each row
single_genre_anime.loc[:, 'genre'] = single_genre_anime['genre'].str.strip()
# Create a set of genres from anime with only one genre
genres_single_genre_anime = set(single_genre_anime['genre'])

# Create a set of all genres found in the DataFrame
all_genres = set(sum(anime['genre'].str.split(', ').tolist(), []))

# Find genres exclusive to multiple genre anime
genres_only_in_multiple_anime = all_genres - genres_single_genre_anime

# Print the genres exclusive to multiple genre anime, if any
if genres_only_in_multiple_anime:
    print("Genres only present in multiple genre anime:")
    for genre in genres_only_in_multiple_anime:
        if genre not in ['Hentai', 'Yuri', 'Yaoi', 'Shounen Ai', 'Shoujo Ai']:
            print(genre)
else:
    print("There are no genres exclusive to multiple genre anime.")
```

```
Genres only present in multiple genre anime:
Police
Super Power
Josei
```

- Some Animes in the dataset have no genres or multiple genres
  - We drop rows without genres from our dataset
  - For Animes with multiple genres, we first created a set of all genres present
  - Then, we find genres exclusive to multi-genre Animes
    - ‘Police’, ‘Superpower’, ‘Josei’

Data Preparation: Filtering Genres

genre	
Action	5.815472
Adventure	6.255949
Cars	5.562000
Comedy	5.852380
Dementia	4.598467
Demons	4.145000
Drama	6.178558
Ecchi	6.218095
Fantasy	5.987636
Game	6.056364
Harem	5.935000
Historical	6.195882
Horror	5.283500
Kids	5.657157
Magic	5.871500
Martial Arts	7.550000
Mecha	5.545556
Military	4.505000
Music	5.756532

- We then calculated the mean rating for the individual genres
- ‘Martial Arts’ has the highest mean rating, reflecting its overall popularity

Data Preparation: Averaging the Averages

```
# Calculate the overall mean rating of all anime_clean
overall_mean_rating = anime_clean['rating'].mean()

# Define a function to replace the genre column with the combined average rating
def replace_genre_with_combined_avg(row):
    genres = row['genre'].split(', ')
    ratings = []
    for genre in genres:
        if genre in mean_rating_by_genre:
            ratings.append(mean_rating_by_genre[genre])
        else:
            ratings.append(overall_mean_rating)
    return sum(ratings) / len(ratings)

# Apply the function to each row in the DataFrame
anime_clean['genre'] = anime_clean.apply(replace_genre_with_combined_avg, axis=1)
anime_clean['genre'].describe()
```

Data Preparation: Ratings for Genre

	anime_id	name	genre	type	episodes	rating
0	32281	Kimi no Na wa.	5.994175	6.334061	1	9.37
1	5114	Fullmetal Alchemist: Brotherhood	5.871699	6.899248	64	9.26
2	28977	Gintama°	5.953125	6.899248	51	9.25
3	9253	Steins;Gate	6.292683	6.899248	24	9.17
4	9969	Gintama'	5.953125	6.899248	51	9.16
...	...	...	...	...	...	...
10891	11095	Zouressha ga Yatte Kita	6.255949	6.334061	1	6.06
10892	7808	Zukkoke Knight: Don De La Mancha	6.077553	6.899248	23	6.47
10893	28543	Zukkoke Sannin-gumi no Hi Asobi Boushi Daisakusen	5.917858	6.464967	1	5.83
10894	18967	Zukkoke Sannin-gumi: Zukkoke Jikuu Bouken	6.041209	6.464967	1	6.13
10895	13455	Zumomo to Nupepe	5.852380	6.899248	32	7.00

10733 rows × 6 columns

- We noted that it is difficult to determine the rating for multi-genre Animes, as each individual genre has its own rating
- Thus, for multi-genre Animes, we replaced the rating with the overall mean rating for all Animes

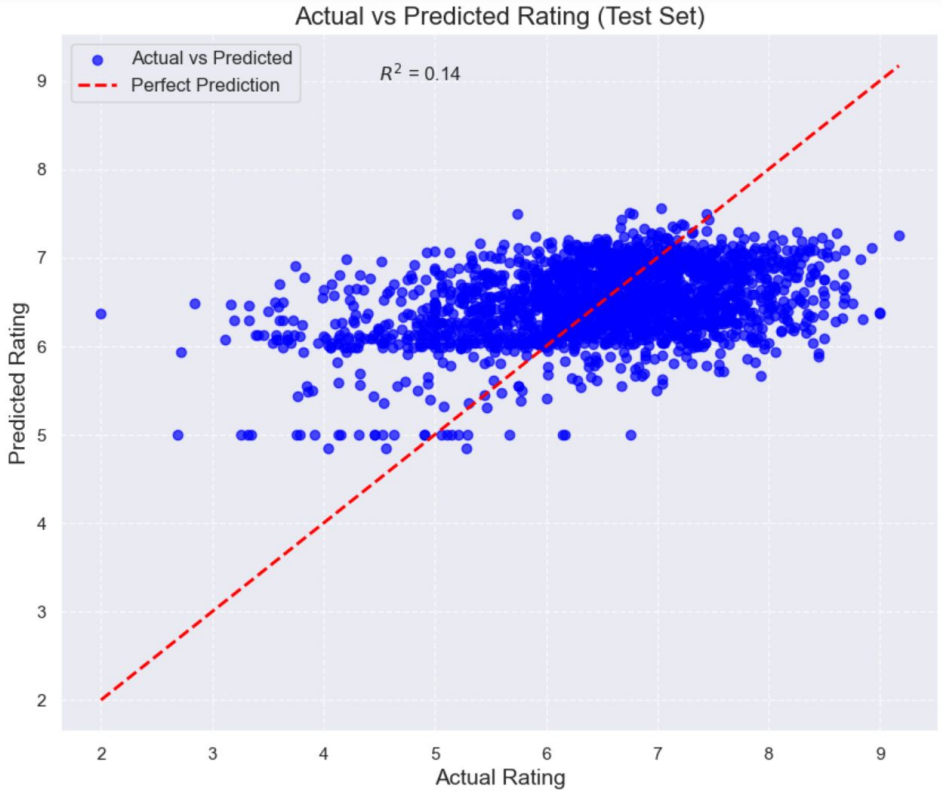
Data Preparation: Ratings for Type

	anime_id	name	genre	type	episodes	rating
0	32281	Kimi no Na wa.	5.994175	6.334061	1	9.37
1	5114	Fullmetal Alchemist: Brotherhood	5.871699	6.899248	64	9.26
2	28977	Gintama°	5.953125	6.899248	51	9.25
3	9253	Steins;Gate	6.292683	6.899248	24	9.17
4	9969	Gintama'	5.953125	6.899248	51	9.16
...	...	...	...	...	...	...
10891	11095	Zouressha ga Yatte Kita	6.255949	6.334061	1	6.06
10892	7808	Zukkoke Knight: Don De La Mancha	6.077553	6.899248	23	6.47
10893	28543	Zukkoke Sannin-gumi no Hi Asobi Boushi Daisakusen	5.917858	6.464967	1	5.83
10894	18967	Zukkoke Sannin-gumi: Zukkoke Jikuu Bouken	6.041209	6.464967	1	6.13
10895	13455	Zumomo to Nupepe	5.852380	6.899248	32	7.00

10733 rows × 6 columns

- We also calculated the mean rating for each type of Anime (TV, Movie, OVA, Others)
  - [6.33 6.90 6.46 6.15] (3 s.f.)

Machine Learning: Linear Regression



We aim to find the relationship between these variables and the ratings of an Anime

- X = ‘Genre’, ‘Type’, ‘Episodes’
- Y = ‘Rating’

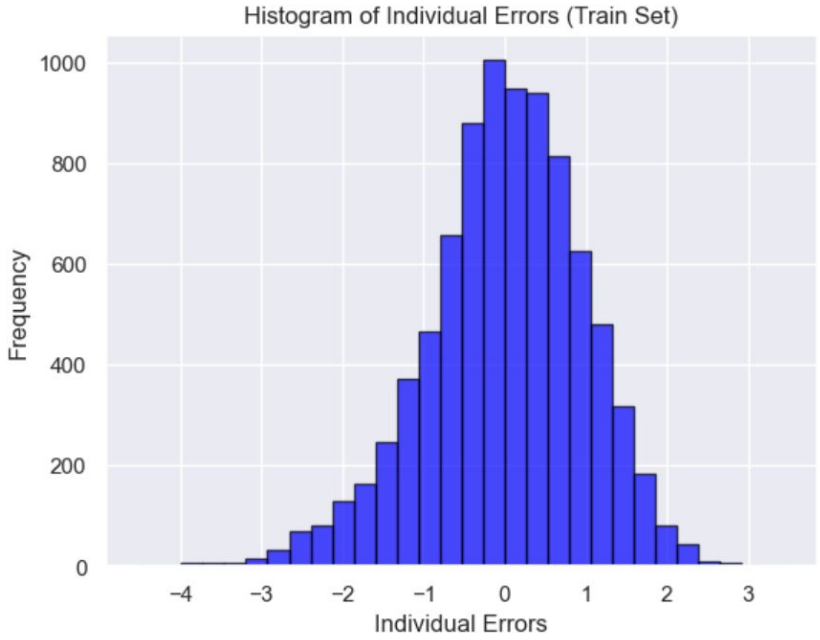
In using a Linear Regression

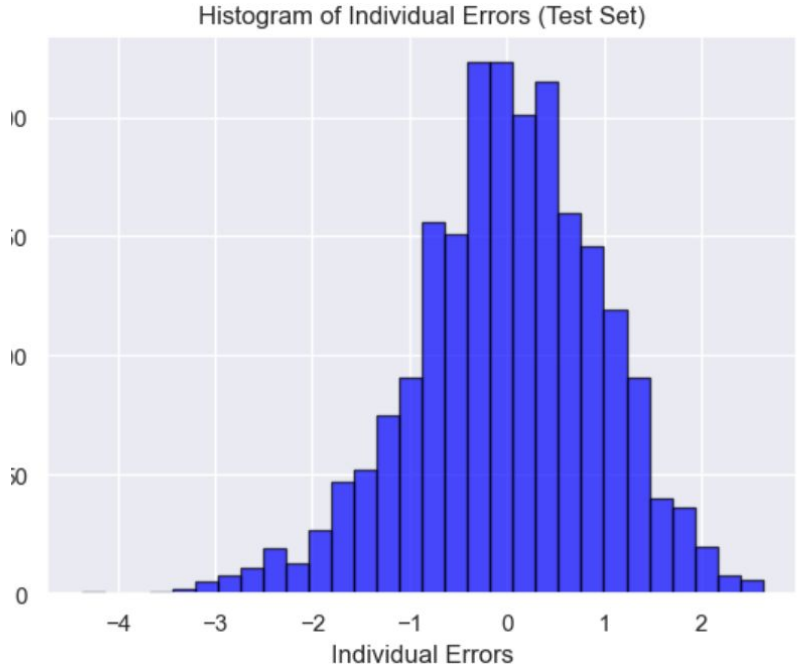
- Split into train (80%) and test (20%) sets
- Create and fit linear regression model on train set
- Make predictions on training set
- Use trained model on test set

Machine Learning: Linear Regression

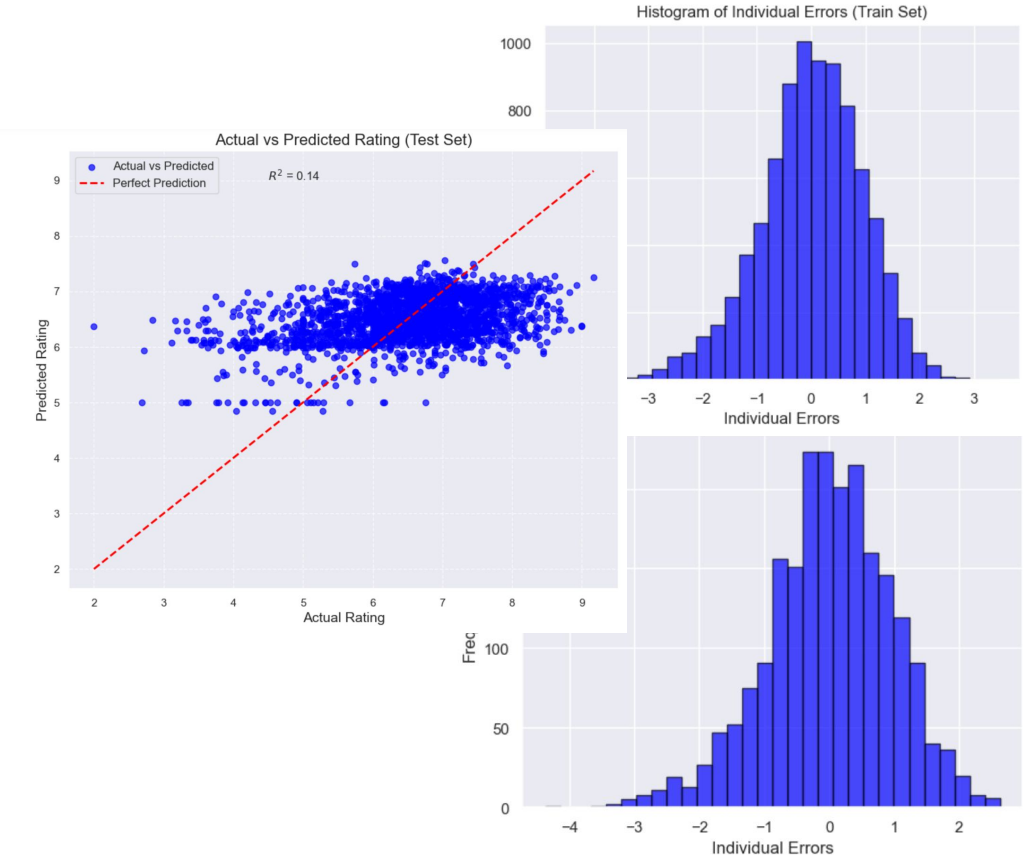
Data	Mean-Squared Error	R^2	Adjusted R^2
Train	0.93391	0.15202	0.15172
Test	0.93625	0.14325	0.14205

Machine Learning: Linear Regression





Machine Learning: Linear Regression



- MSE for both train and test sets are less than 1, despite the large variation as seen in the linear regression graph

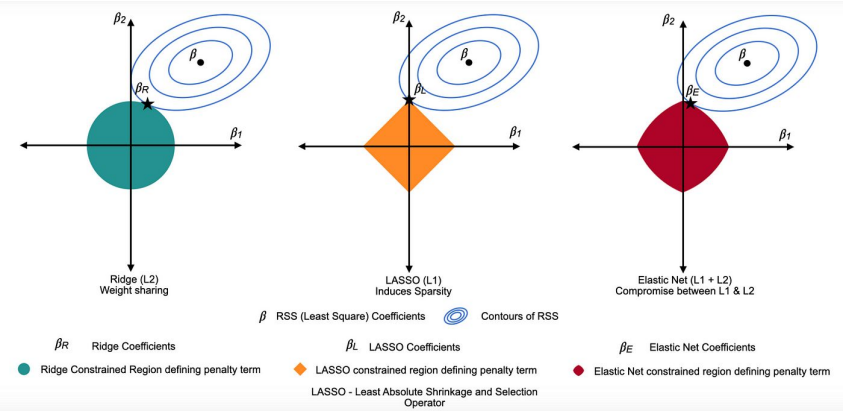
- This could be because most individual errors are of value -1 < e < 1, so the MSE would be a value smaller than 1

Machine Learning: Linear Regression

Variable	Variable Coefficient (5 s.f.)
Genre	1.0277
Type	0.90942
Episodes	-0.000089000
Intercept	-5.4824

The intercept of -5.48 might seem alarming, but this is due to how our variables are handled

Machine Learning: Lasso, Ridge, and Elastic Net Summary



- Given our large dataset (>10k values), using these regressions could help prevent overfitting in our linear regression model as a penalty term is added to the loss function
- Using scikit learn, we are able to implement these with hyperparameter tuning and cross validation

Data Preparation: One-Hot Encoding on ‘Genre’ column

	anime_id	name	type	episodes	rating	Action	Adventure	Cars	Comedy	Dementia	...	Seinen	Shoujo	Shounen	Slice of Life	Space	Sports
0	32281	Kimi no Na wa.	Movie	1	9.37	0	0	0	0	0	...	0	0	0	0	0	0
1	5114	Fullmetal Alchemist: Brotherhood	TV	64	9.26	1	1	0	0	0	...	0	0	1	0	0	0
2	28977	Gintama°	TV	51	9.25	1	0	0	1	0	...	0	0	1	0	0	0
3	9253	Steins;Gate	TV	24	9.17	0	0	0	0	0	...	0	0	0	0	0	0
4	9969	Gintama'	TV	51	9.16	1	0	0	1	0	...	0	0	1	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
10891	11095	Zouressha ga Yatte Kita	Movie	1	6.06	0	1	0	0	0	...	0	0	0	0	0	0
10892	7808	Zukkoke Knight: Don De La Mancha	TV	23	6.47	0	1	0	1	0	...	0	0	0	0	0	0
10893	28543	Zukkoke Sannin-gumi no Hi Asobi Boushi Daisakusen	OVA	1	5.83	0	0	0	0	0	...	0	0	0	0	0	0
10894	18967	Zukkoke Sannin-gumi: Zukkoke Jikuu Bouken	OVA	1	6.13	0	0	0	1	0	...	0	0	0	0	0	0
10895	13455	Zumomo to Nupepe	TV	32	7.00	0	0	0	1	0	...	0	0	0	0	0	0

Data Preparation: One Hot Encoding of Column ‘Type’

pisodes	rating	Action	Adventure	Cars	Comedy	Dementia	Demons	...	Space	Sports	Super Power	Supernatural	Thriller	Vampire	Movie	OVA	Others	TV
1	9.37	0	0	0	0	0	0	...	0	0	0	1	0	0	True	False	False	False
64	9.26	1	1	0	0	0	0	...	0	0	0	0	0	0	False	False	False	True
51	9.25	1	0	0	1	0	0	...	0	0	0	0	0	0	False	False	False	True
24	9.17	0	0	0	0	0	0	...	0	0	0	0	1	0	False	False	False	True
51	9.16	1	0	0	1	0	0	...	0	0	0	0	0	0	False	False	False	True
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1	6.06	0	1	0	0	0	0	...	0	0	0	0	0	0	True	False	False	False
23	6.47	0	1	0	1	0	0	...	0	0	0	0	0	0	False	False	False	True
1	5.83	0	0	0	0	0	0	...	0	0	0	0	0	0	False	True	False	False
1	6.13	0	0	0	1	0	0	...	0	0	0	0	0	0	False	True	False	False
32	7.00	0	0	0	1	0	0	...	0	0	0	0	0	0	False	False	False	True

Data Preparation: Converting True/False in ‘Type’ to ‘1’ or ‘0’

episodes	rating	Action	Adventure	Cars	Comedy	Dementia	Demons	...	Space	Sports	Super Power	Supernatural	Thriller	Vampire	Movie	OVA	Others	TV
1	9.37	0	0	0	0	0	0	...	0	0	0	1	0	0	1	0	0	0
64	9.26	1	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1
51	9.25	1	0	0	1	0	0	...	0	0	0	0	0	0	0	0	0	1
24	9.17	0	0	0	0	0	0	...	0	0	0	0	1	0	0	0	0	1
51	9.16	1	0	0	1	0	0	...	0	0	0	0	0	0	0	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1	6.06	0	1	0	0	0	0	...	0	0	0	0	0	0	1	0	0	0
23	6.47	0	1	0	1	0	0	...	0	0	0	0	0	0	0	0	0	1
1	5.83	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	0	0
1	6.13	0	0	0	1	0	0	...	0	0	0	0	0	0	0	1	0	0
32	7.00	0	0	0	1	0	0	...	0	0	0	0	0	0	0	0	0	1

Machine Learning: Lasso, Ridge, and Elastic Net

```
#Define the parameter grid for Lasso, Ridge, and ElasticNet
lasso_param_grid = {'alpha': [0.001, 0.01, 0.1, 1, 10]}
ridge_param_grid = {'alpha': [0.001, 0.01, 0.1, 1, 10]}
elastic_net_param_grid = {'alpha': [0.001, 0.01, 0.1, 1, 10],
                          'l1_ratio': [0.1, 0.3, 0.5, 0.7, 0.9]}
```

```
# Perform GridSearchCV for Lasso
lasso_grid_search = GridSearchCV(lasso_model, param_grid=lasso_param_grid, cv=5, scoring='neg_mean_squared_error')
lasso_grid_search.fit(x_train, y_train)

# Perform GridSearchCV for Ridge
ridge_grid_search = GridSearchCV(ridge_model, param_grid=ridge_param_grid, cv=5, scoring='neg_mean_squared_error')
ridge_grid_search.fit(x_train, y_train)

# Perform GridSearchCV for ElasticNet
elastic_net_grid_search = GridSearchCV(elastic_net_model, param_grid=elastic_net_param_grid, cv=5, scoring='neg_mean_squared_erro
elastic_net_grid_search.fit(x_train, y_train)
```

Machine Learning: Lasso, Ridge, and Elastic Net

```
# Get best parameters and best scores for Lasso
best_params_lasso = lasso_grid_search.best_params_
best_score_lasso = lasso_grid_search.best_score_

# Get best parameters and best scores for Ridge
best_params_ridge = ridge_grid_search.best_params_
best_score_ridge = ridge_grid_search.best_score_

# Get best parameters and best scores for ElasticNet
best_params_elastic_net = elastic_net_grid_search.best_params_
best_score_elastic_net = elastic_net_grid_search.best_score_
```

```
# Fit models with best parameters
lasso_model_best = Lasso(alpha=best_params_lasso['alpha'])
ridge_model_best = Ridge(alpha=best_params_ridge['alpha'])
elastic_net_model_best = ElasticNet(alpha=best_params_elastic_net['alpha'], l1_ratio=best_params_elastic_net['l1_rat

lasso_model_best.fit(x_train, y_train)
ridge_model_best.fit(x_train, y_train)
elastic_net_model_best.fit(x_train, y_train)
```

Analysis: Lasso, Ridge, and Elastic Net

Regression Model	Best Parameters ('alpha')	Best Score (Negative MSE)	R^2	Adjusted R^2
Lasso	0.001	0.75430	0.28797	0.27341
Ridge	1	0.75375	0.28515	0.27053
Elastic Net	0.001 L1 ratio: 0.1	0.75373	0.28634	0.27175

Analysis

Lasso Coefficients:

	Variable	Coefficient
14	Josei	0.613858
21	Mystery	0.493767
7	Drama	0.477783
31	Shounen	0.444997
19	Military	0.417975
24	Psychological	0.365998
37	Thriller	0.355348
27	School	0.350816
23	Police	0.341608
42	TV	0.328801
36	Supernatural	0.327427
29	Seinen	0.321258
25	Romance	0.278351
12	Historical	0.251644
9	Fantasy	0.238845
32	Slice of Life	0.215026
30	Shoujo	0.208354
10	Game	0.107170
26	Samurai	0.103836
16	Magic	0.072987
11	Harem	0.030817
0	episodes	0.000430
8	Ecchi	-0.000000
3	Cars	0.000000
6	Demons	0.000000
38	Vampire	0.000000
39	Movie	0.000000
17	Martial Arts	0.000000
40	OVA	-0.008339
33	Space	-0.032093
18	Mecha	-0.034232
20	Music	-0.057718
41	Others	-0.097197
15	Kids	-0.178364
13	Horror	-0.180306
5	Dementia	-0.910288

Lasso coefficients (a sample from total printed; incomplete)

- We printed the corresponding coefficients for Lasso, Ridge, and Elastic Net in a descending order

Analysis

Regression Model	Best Parameters ('alpha')	Best Score (Negative MSE)	R^2	Adjusted R^2
Lasso	0.001	0.75430	0.28797	0.27341

- Lasso tends to do well if there are a small number of significant parameters and the others are close to zero (when only a few predictors actually influence the response)

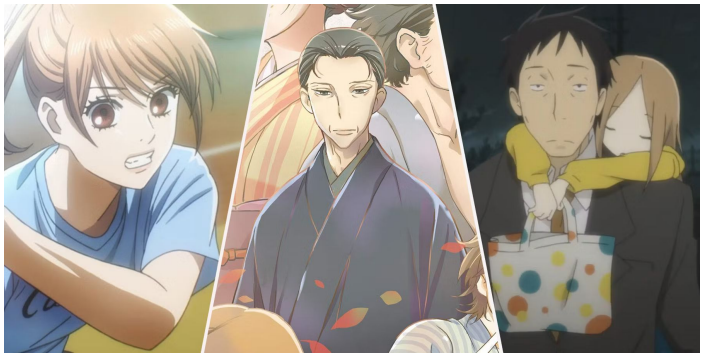
Analysis

Lasso Coefficients:

	Variable	Coefficient
14	Josei	0.613858
21	Mystery	0.493767
7	Drama	0.477783
31	Shounen	0.444997
19	Military	0.417975
24	Psychological	0.365998
37	Thriller	0.355348
27	School	0.350816
23	Police	0.341608
42	TV	0.328801
36	Supernatural	0.327427
29	Seinen	0.321258
25	Romance	0.278351
12	Historical	0.251644
9	Fantasy	0.238845
32	Slice of Life	0.215026
30	Shoujo	0.208354

- Josei gives the highest positive coefficient
- Greatest association with ratings
- Thus, an Anime with the ‘Josei’ genre tends to have the greatest positive influence on its ratings - it perhaps matches with audiences’ preferences

Outcome



- 'Martial Arts' and ‘TV’ has the highest mean rating
- From Lasso regression, 'Josei' has the highest positive coefficient, whereas ‘Martial Arts’ is assigned a coefficient of 0.
    - This could be due to a difference in rating distribution of ‘Martial Arts’ anime with one and multiple genres
- These analysis gave us an insight on the preferred genres of the general public

Evaluation

- We acknowledge that there may be other factors affecting the rating of an Anime (e.g. art style)
  - However, we also note that these factors may be difficult to quantify for use as predictors
- Ratings in our datasets may be skewed as Anime with large communities may be more favourably reviewed

Evaluation

- Our model only predicts ratings between ~4.8 - 7.7, so it does not predict the rating of outliers well
  - There are many outlier entries and we cannot simply remove them from the analysed data

Conclusion

- Initially, we thought that renowned genres like ‘Action’ or ‘Adventure’ would be better received and thus have higher ratings
- However, given their popularity, the anime within these genres may have a wider spread due its differing quality

Conclusion

- Instead, the smaller number of ‘Martial Arts’ anime leads to a more niche community with less spread, perhaps indicating a higher quality of Anime
- Inclusion of other genres that could increase the rating (based on regression analysis), such as ‘Josei’, ‘Mystery’, and ‘Drama’ may help boost the ratings of a certain Anime

Conclusion

Based on our analysis, a new Anime, without any existing fanbase or community, is most likely to have a higher rating if it is a

- **TV show**
- **Includes ‘Martial Arts’, ‘Josei’, ‘Mystery’, ‘Drama’, as its genres**