Data Processing Final Project Report
Predicting Anime Popularity from other Data

Ilaria Zanini

The purpose of this research is to determine whether the variables of scores (how good an anime is rated), number of episodes and genres can predict how popular an anime is. For these purposes 3 linear regressions will be conducted to determine whether each model is effective. The research question is:

Can an anime's score, number of episodes or type of genre be used to predict how popular it is?

For each model similar statistical analysis were used to allow the comparison of the models between each other. The mean squared error gives the average of squared errors between estimates and the actual values, which indicates how wrong the predictions are on average. R^2 tells us how much variance in the data the variable explains. The closer R^2 is to 1, the more the variable explains the data. R^2 can also be adjusted to account for more variables, which can cause overfitting.

Score and Popularity

Before beginning the analysis, for the purpose of visibility the data of the popularity column was transformed so that it would range from 0 to 10.5 instead of 0 to 1000. To run a linear regression, 80% of the data was used for training the model, while 20% was kept as data that the model never saw before. The purpose of splitting and testing the data compared to doing a normal linear regression is so that the model can be tested on data it never saw before. The train_test_split function splits the data into 4 parts which will be used for the linear regression. The following 4 parts are:

1. X_train_score: 80% of the anime which are used to train the model
2. X_test_Score: 20% of the anime are used to test the model
3. Y_train_score: the popularity values of the training anime
4. Y_test_score: the popularity values of the testing anime

During this analysis and all the following ones, the "random_state" function will be used to set a seed for replicability purposes.

After splitting the data, the function LinearRegression() from sklearn allows to automatically fit the model. The fit function trains the model and finds the line of best fit which is based on the formula:

$$y = mx + b$$

Where m is the slope coefficient, b is y intercept, y is the predicted value and x is the independent variable.

Then the model is used to predict popularity.

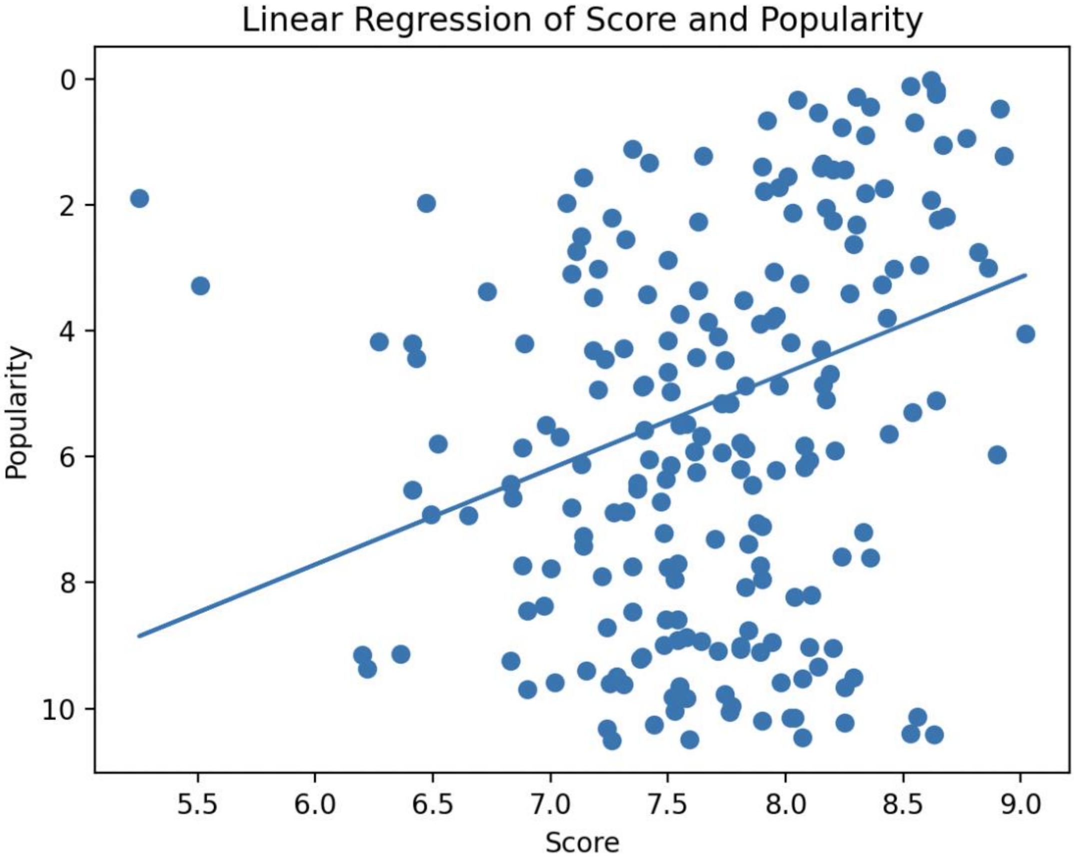


Figure 1. The plotting of the linear regression.

Figure 1 indicates a relationship between score and popularity, which is visible from the fact that the line is not flat. However, to determine the strength and significance of the relationship statistical tests will be used.

The calculation of the Standard Error was calculated in the following way:

1. First the residuals were obtained. Those are the actual training values for y minus the predictions for y. For a visual explanation, please refer to figure 2.

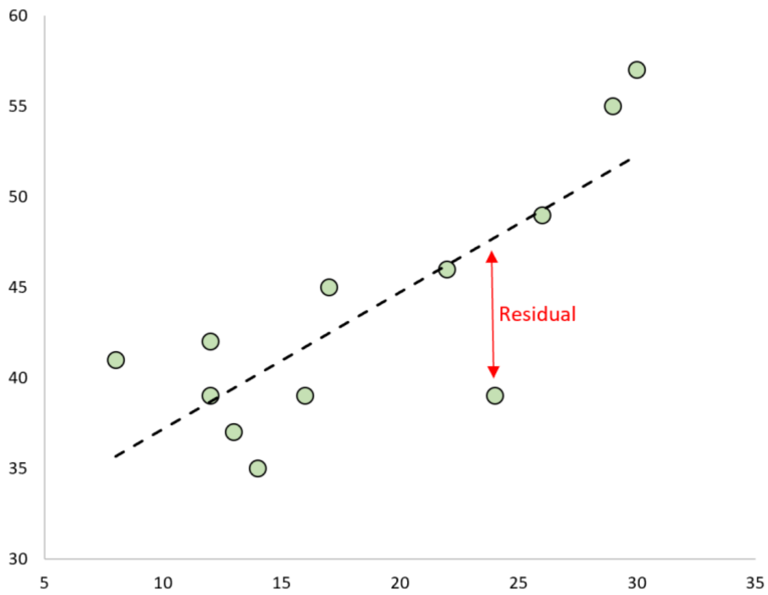


Figure 2. Bobbitt, 2020

2. The residual sum of squares (RSS) was calculated by summing all residuals and squaring them
3. Sigma squared was obtained by dividing the RSS by the number of rows minus 2 (the degrees of freedom)
4. Sum of squares of X (SSx) was calculated by subtracting x (from the training set) with its mean, and summing and squaring those differences
5. Finally, the Standard Error was obtained with the following formula

$$SE = \sqrt{\sigma^2 / SSx}$$

To further evaluate the model the R^2 and the Mean Squared Error were obtained with functions from the sklearn package. The adjusted R^2 calculates a more accurate value of R^2 since it penalizes the model for adding extra variables. While not relevant here since the model only has one variable, it will be relevant for comparing this model to the next ones.

To determine whether the relationship between score and popularity is significant a double-sided t-test will be used. The t statistic was calculated by dividing the model's coefficient by the Standard Error. Then the p-value was calculated through a function from the scipy package. The null hypothesis is that there is no relationship between score and popularity (the slope would be equal to 0), while the alternative hypothesis is that there is a relationship between score and popularity. A p-value lower than 0.05 would support the alternative hypothesis.

To visualize the data further, a decision tree was made. Each branch of the decision tree splits the data by score. For example, firstly the anime that have a score below 7.7 are allocated to the left side of the tree. The model divides the data into different score ranges. The value on the decision tree is the predicted popularity for anime in that node of the tree. On the final line, it is visible that the nodes with a larger sample have a lower value, which means that anime with lower score are more popular. For the purpose of interpretability, the depth of the tree was limited to 3, otherwise the plot would be too cluttered to interpret it.

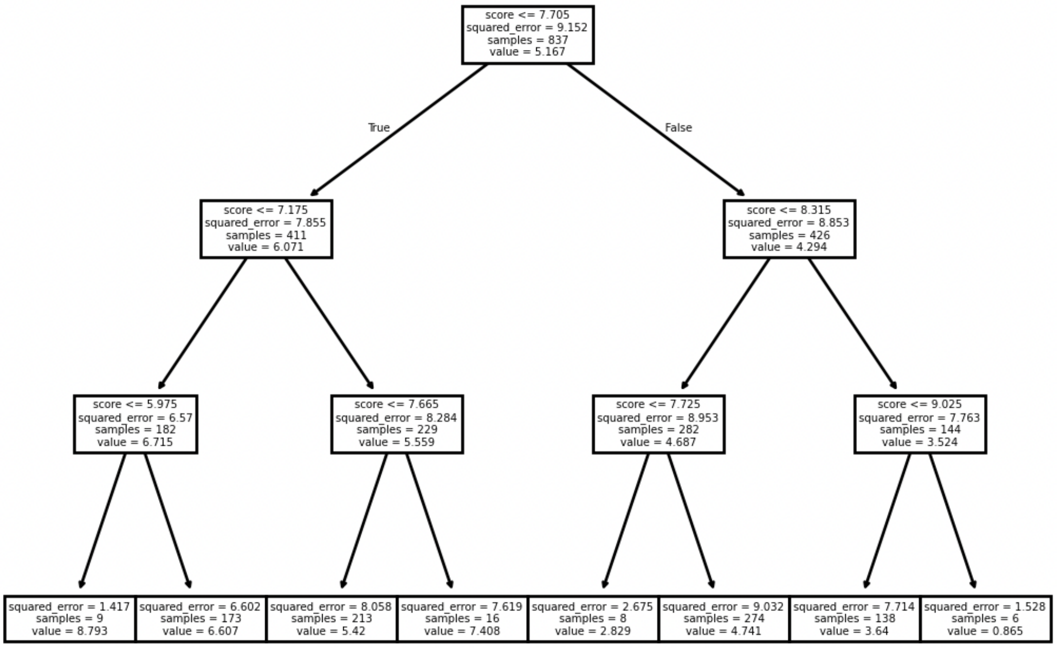


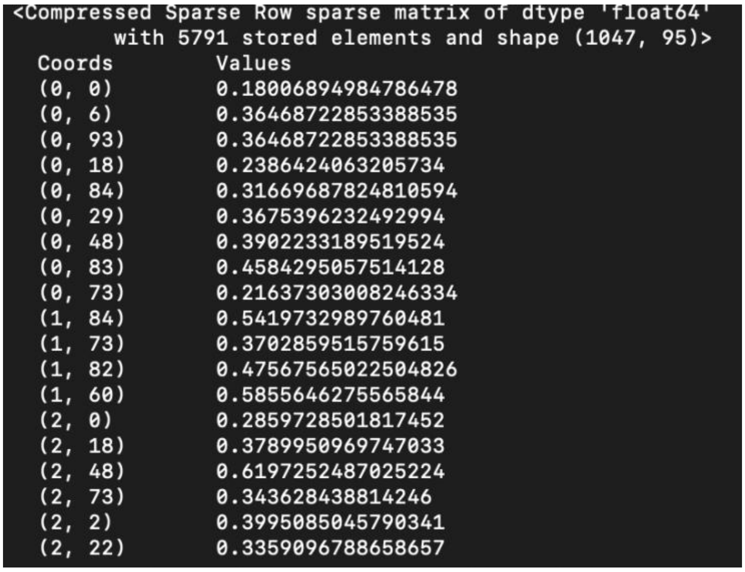
Figure 3. Decision tree diagram

The results from all the analysis are the following:
Mean Standard Error: 8.896
R^2: 0.027
Adjusted R^2: 0.022
Standard Error: 0.146
P-value: 0.0

The results are mixed. A high mean standard error and a R^2 close to 0 indicate a poor performance of predicting popularity by the model. A R^2 close to 1 would indicate a linear relationship between score and popularity, but the result is very close to 0. On the other hand, the Standard Error is low compared to the range of popularity, which indicates that the estimate of the slope is reliable, and a p-value of 0 indicates a significant relationship between score and popularity. The results of the model overall indicate that while there is a significant relationship between score and popularity, the relationship is not strong enough to be able to predict popularity from score.

Genres and Popularity

Another linear regression will be used to determine whether the genres of an anime can be used to predict its popularity. As the previous linear regression, the data for popularity was modified so that it would be a more readable range. Since genres is a variable in text form, TF-IDF will be used. TF-IDF will be used to determine how important that word is for the current anime, compared to other anime. Using the TfidVectorizer() function from the sklearn package, a vectorizer was initiated and then it computed a TF-IDF score for every word.



```
<Compressed Sparse Row sparse matrix of dtype 'float64'
        with 5791 stored elements and shape (1047, 95)>
  Coords        Values
  (0, 0)        0.18006894984786478
  (0, 6)        0.36468722853388535
  (0, 93)       0.36468722853388535
  (0, 18)       0.2386424063205734
  (0, 84)       0.31669687824810594
  (0, 29)       0.3675396232492994
  (0, 48)       0.3902233189519524
  (0, 83)       0.4584295057514128
  (0, 73)       0.21637303008246334
  (1, 84)       0.5419732989760481
  (1, 73)       0.3702859515759615
  (1, 82)       0.47567565022504826
  (1, 60)       0.5855646275565844
  (2, 0)        0.2859728501817452
  (2, 18)       0.3789950969747033
  (2, 48)       0.6197252487025224
  (2, 73)       0.343628438814246
  (2, 2)        0.3995085045790341
  (2, 22)       0.3359096788658657
```

Figure 4. The TF-IDF scores matrix.

The numbers in score represent the anime ID, word ID and then are followed by the value for that word. A linear regression model was trained as done in the previous model. However, due to the different structure of the data (the matrix has 95 features), a linear regression graph could not be plotted. The data was visualized with a cluster graph. The clusters will be based on anime with similar genres description (determined from the TF-IDF score).

To determine the best number of clusters I will use the elbow method. With a loop I computed the inertia (the clustering error of all points in the cluster) for different possible clustering amounts. The elbow diagram is the following:

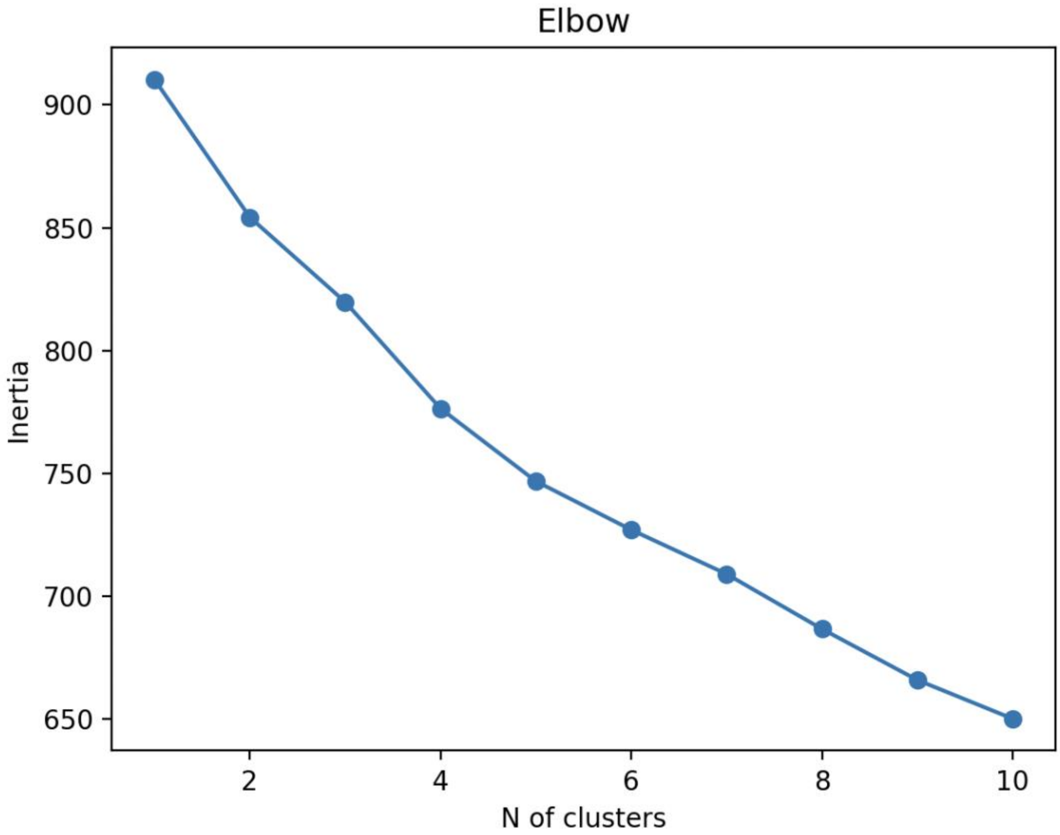


Figure 5. Elbow diagram

While in the diagram there is no sharp elbow, the decrease tends to flatten after 3 points, therefore 3 clusters will be used for the rest of the analysis.

To determine the average popularity of each cluster, a KMeans model was fitted, and each anime was assigned to a cluster. After each anime was assigned to a cluster, the mean popularity was calculated for each cluster. The results were the following:
Cluster 0: 5.236
Cluster 1: 5.302
Cluster 2: 5.297

Next an Anova was run to determine whether there were significant differences between the clusters. Anova is a statistical analysis used to determine whether several means differ from each other. With the pandas groupby function I grouped the popularity values per cluster. An Anova was run with the automatic function f_oneway function.

The null hypothesis of the ANOVA is:

$$H0:\ \mu1 = u2 = \ \mu3$$

The alternative hypothesis is:

$$Ha: \mu1 \neq u2 \neq \ \mu3$$

If the p-value will be less than 0.05, it will be possible to reject H0 and determine that there is a difference between the cluster means.

To determine the names of the clusters, firstly the TF-IDF matrix was converted to a data frame. Then the average score per genre in the cluster was obtained and then the name of the genre with the highest TF-IDF score was returned. Cluster 0 was the cluster of genre “school”, cluster 1 of genre “fantasy” and cluster 2 of genre “isekai”.

Clustering gave the possibility of analyzing the differences between cluster means, however principal component analysis (PCA) is still needed to visualize the data. The data is currently at 95 dimensions. PCA allows us to reduce the dimensions to 2. Firstly, a PCA model was created, and the data was fitted to the model and the graph of the two components that explained the most patterns in the data were plotted.

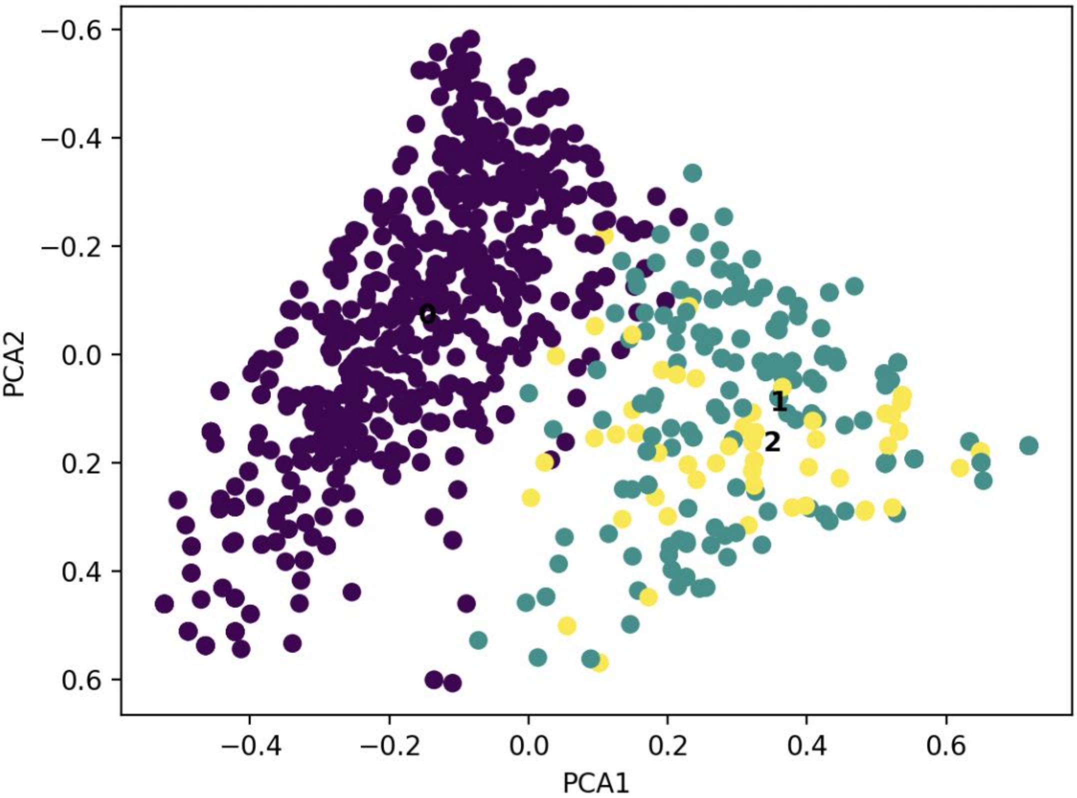


Figure 5. Graph from the PCA analysis. Cluster 0 is school, cluster 1 is fantasy and cluster 2 is isekai.

The results of the analysis are the following:

Mean squared error: 9.099
R^2: 0.004
Adjusted R^2: -0.824
F-statistic: 0.052
P-value: 0.949

Compared to the range of the data, the squared error is very high, and the R^2 is very low, indicating that the predictions of the models are not accurate. However, the adjusted R^2 was also calculated. The adjusted R^2 penalizes the model for adding more predictors, so it avoids overfitting. After the adjustments, the model performs worse, even below guessing levels. The p-value above 0.05 indicates that there is no statistically significant difference between cluster means. The F-statistic is the ratio of variance between groups, and the low F-statistic of 0.052 indicates that any differences in the means are random.

In conclusion, this model performs poorly. The type of genres does not predict popularity. In the clustering diagram, clusters 1 and 2 are overlapped, and even if cluster 0 differs the other tests determined that the difference was not significant.

Score, episodes and Popularity

A linear regression was done to see if the number of episodes and score could predict an anime's popularity. Similar steps to the first linear analysis were done. The results of the analysis are the following:

Mse: 8.77

R^2: 0.041

Adjusted R^2: 0.032

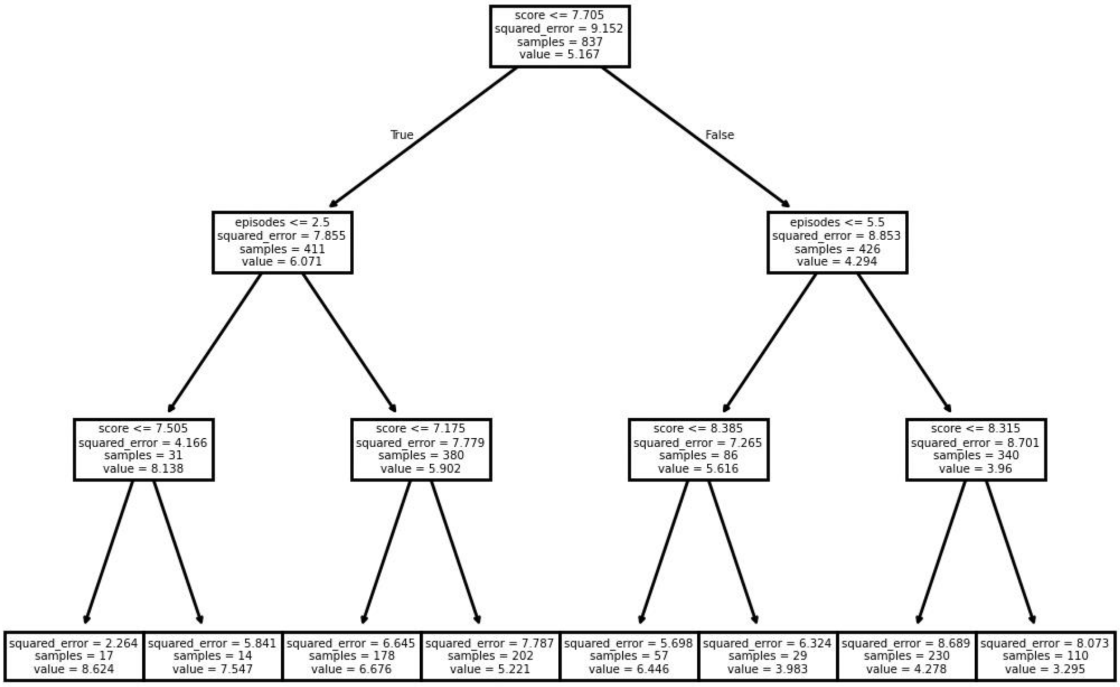


Figure 6: decision tree

The decision tree initially shows the same pattern as the previous decision tree (Figure 3), where anime with a lower score are more popular. The decision tree shows splits both for score and number of episodes; in each node the anime with less episodes are more popular.

Overall, the results for this model are also poor. The mse suggests a lot of error from the model and the low R^2 tells that the variables score and number of episodes do not explain any variance in popularity.

Conclusion

None of the models have succeeded in predicting popularity, particularly the second and third linear regressions. The linear regression for genres and popularity did not produce any significant statistical results meaning the different clusters of genres were not different. The only linear regression that produced a significant result is the linear regression for score and popularity. The t-test produced a significant result, meaning that the slope is significantly different from 0, indicating a relationship between score and popularity. While there is a significant relationship, the model still does not predict popularity. A reason for why the model could have failed was that popularity was a ranked variable (from 1 to 1050), meaning that there was only a limited number of high-popularity anime. The structure to the data could have contributed to the poor results.

References

Bobbitt, Z. (2020, December 7). *What are residuals in statistics?* Statology. https://www.statology.org/residuals/

Appendix.

Output of popularity.py:

```
Intercept: 16.829050371493405
Coefficients: [-1.51965607]
     score
519    7.51
411    8.38
538    8.26
229    7.72
602    7.73
...     ...
491    8.38
257    7.18
1035   7.25
736    7.84
528    7.79

[837 rows x 1 columns]
SE: 0.1461192293949787
```

```
mse 8.895870227665364
r2 0.02712168821613936
r2 adjusted 0.02244438864025544
pvalue [0.]
```

Output if popularity_genres.py

```
Coefficients: [-1.69833928 -0.29095751 -1.58545416  0.58749083  0.4502234  -0.61531279
 -1.49133309  1.84570154 -0.29095751 -0.31457516 -0.80572462  0.81522085
 -0.47529246 -0.33211937 -1.60198505 -0.53323107 -1.32049436 -0.25957593
 -1.142947   -0.12706729 -4.60915659  1.49723703  2.41355062  1.13997405
  0.42577063 -0.23870388 -0.79317326 -0.61531279  3.12941963 -2.64880892
 -1.10641196  1.49295635  0.8808431  -0.30322062  0.38400858 -0.23870388
  1.13997405 -1.79920562  4.44959012  2.57429227  1.54105321 -1.32346672
  1.66040049  2.71751225  3.08037645 -1.88072033 -0.30869214  0.36489507
 -0.0567806  -0.27706273  0.47588914 -0.79643673 -1.32346672 -0.33211937
 -0.53323107  0.43282355  1.86565889  4.15136756 -1.54036288 -0.39629319
 -3.64343713 -0.00717199 -0.53371365  0.44542346  1.92042618 -0.0078696
 -1.56606461 -1.48270572  0.42577063  0.659188    2.71751225  2.71751225
 -1.55983895 -0.52246098  1.90571443 -1.32346672  2.79407866  0.10744951
  0.8808431  -0.00717199 -1.9317438  -0.39629319 -0.0364153  -3.06552545
  1.0394926  -3.11317058 -1.01912379 -1.01912379 -3.84683103 -1.41522205
 -0.07167173  2.1070447   0.89226365 -1.49133309  2.36481138]
mse 9.09900470578105
r2 0.004906309273257747
r2 adjusted -0.8243384329990275
cluster
0    5.235927
1    5.302458
2    5.296667
Name: pop_mod, dtype: float64
F 0.051886262614448085
```

```
P 0.9494392953400622
['action' 'adult' 'adventure' 'anthropomorphic' 'arts' 'avant' 'award'
 'boys' 'cast' 'cgdct' 'childcare' 'combat' 'comedy' 'crime'
 'crossdressing' 'culture' 'delinquents' 'detective' 'drama' 'ecchi'
 'educational' 'erotica' 'fantasy' 'female' 'fi' 'gag' 'game' 'garde'
 'girls' 'gore' 'gourmet' 'harem' 'high' 'historical' 'horror' 'humor'
 'idols' 'isekai' 'iyashikei' 'josei' 'kids' 'life' 'love' 'magical'
 'mahou' 'martial' 'mecha' 'medical' 'military' 'music' 'mystery'
 'mythology' 'of' 'organized' 'otaku' 'parody' 'performing' 'pets'
 'polygon' 'power' 'psychological' 'quo' 'racing' 'reincarnation'
 'reverse' 'romance' 'samurai' 'school' 'sci' 'seinen' 'sex' 'shift'
 'shoujo' 'shounen' 'showbiz' 'slice' 'space' 'sports' 'stakes' 'status'
 'strategy' 'super' 'supernatural' 'survival' 'suspense' 'team' 'time'
 'travel' 'urban' 'vampire' 'video' 'villainess' 'visual' 'winning'
 'workplace']
        action     adult  adventure  ...  visual   winning  workplace
0     0.180069  0.000000   0.000000  ...     0.0  0.364687   0.000000
1     0.000000  0.000000   0.000000  ...     0.0  0.000000   0.000000
2     0.285973  0.000000   0.399509  ...     0.0  0.000000   0.000000
3     0.198358  0.359862   0.000000  ...     0.0  0.000000   0.000000
4     0.238372  0.000000   0.000000  ...     0.0  0.482767   0.000000
...        ...       ...        ...  ...     ...       ...        ...
1042  0.206335  0.000000   0.000000  ...     0.0  0.000000   0.000000
1043  0.000000  0.000000   0.000000  ...     0.0  0.000000   0.000000
1044  0.000000  0.000000   0.000000  ...     0.0  0.000000   0.459924
1045  0.000000  0.000000   0.000000  ...     0.0  0.000000   0.000000
1046  0.000000  0.000000   0.000000  ...     0.0  0.000000   0.000000

[1047 rows x 95 columns]
Cluster 0 school
Cluster 1 fantasy
Cluster 2 isekai
```

```
(1047, 2)
[[-0.15815785 -0.05682438]
 [ 0.34631768  0.10314467]
 [ 0.33647709  0.17764843]]
```

Output of popularity_scores_episodes.py

```
Intercept: 16.690832567692325
Coefficients: [-1.46820489 -0.01403141]
mse 8.770217561620191
r2 0.04086343022503236
r2 adjusted 0.031596410227206606
```