Term Project

Final Submission

Nathanael I. Ochoa

Bellevue University

DSC 550: Data Mining

Dr. Brett Werner

June 1, 2024

My business problem/idea involves utilizing the MyAnimeList Anime and Manga Datasets available on Kaggle to determine which manga are worth animating. I understand that this is not the typical process in the real world, so please bear with me. However, I find it intriguing to imagine a scenario where a group of executives sits around a circular table, deciding which manga deserves its own anime adaptation.

The manga/anime industry has maintained its popularity for many years, especially with new advancements in technology, which make reading manga online and streaming anime more accessible. Given that there are significantly more manga series than anime series, it is logical that not all manga have been adapted into anime. Creating new anime projects requires both time and money. Hence, my goal is to develop an analytical model that can utilize a manga's rating to predict the rating it might receive in its anime format. These predictions can assist decision-makers in determining whether resources should be allocated to the creation of new anime projects. When a business suffers losses, it impacts everyone involved, so it is crucial to develop and test a highly accurate model to support these critical decisions.

Here is my pitch to the fictional executives: 'In the rapidly evolving landscape of the anime and manga industry, the challenge of determining which manga series merit adaptation into anime format looms large. With resources at a premium, making informed decisions becomes paramount. We propose a solution: harnessing the power of data analytics to develop a predictive model that estimates the potential rating of manga adaptations. By leveraging data retrieved from MyAnimeList, we can provide you with actionable insights to optimize resource allocation, minimize financial risks, and gain a competitive edge in the market. Our implementation plan involves rigorous testing to ensure accuracy and reliability. Embracing data

analytics is not just about mitigating risks—it is about seizing opportunities and staying ahead of the curve in this dynamic industry landscape'.

MyAnimeList is an online anime and manga community and database. The author of the dataset used APIs to scrape the data from MyAnimeList themselves. Unfortunately, I have no access to profit data for any of the anime and/or manga in the data files I possess, but I will still be able to compare the ratings. I plan on obtaining the anime score data and adding it to the manga dataset. This will be the 'target' variable in the model that I will create.

The manga dataset initially consisted of 64,833 rows and 30 columns, while the anime dataset had 24,985 rows and 39 columns. However, these figures quickly changed during the data preparation phase. At the start, I faced a frustrating obstacle: each dataset had its own 'id' column, but the id numbers were not synchronized. As a result, 'Manga X' and its anime adaptation did not share corresponding id numbers. Initially, I tried using the 'title' columns to link the datasets, only to encounter another challenge. While the manga dataset neatly listed every series in its manga format, the anime dataset, focused on television adaptations, presented series split into seasons. For example, the anime dataset included entries like 'Manga X: Season One', 'Manga X: The Second Season', and 'Manga X: The Final Season', each with distinct id numbers. This complicated the straightforward linkage of entries within the same series. To address this, I created a function that calculates the average score of anime titles containing a specified name, being the manga title. It begins by checking if the input name is valid. If valid, it searches for anime titles containing the name and computes their average score. If no titles match the name, it returns 'no mean'. If the input name is empty or contains only whitespace characters, it returns 'invalid'. This function is invoked using the apply() function and adds a new column to the manga dataset, enabling comparison between the manga score to the 'average anime score'

within the same dataset. Additionally, I developed a similar function to compute the average of the 'favorites' column, indicating the number of users who favorited the anime. This resulted in the creation of a new column labeled 'average anime favorites'. I then created a new dataframe that only contained the following variables: 'title[1]', 'title_english[2]', 'type[3]', 'status[4]', 'chapters[5]', 'sfw[6]', 'score[7]', 'avg_anime_score', 'favorites[8]', and 'avg_anime_favs'. Additionally, I generated dummy variables from the 'genre[9]' column and merged the results into the new dataframe.

Following this, I addressed any missing values in the 'title_enlish' column by replacing them with the term 'unavailable'. Furthermore, instances of 'no mean' values were substituted with the corresponding row's 'score' value, while any null values in the 'score' column were replaced with the 'avg_anime_score' value. Rows containing null values in both score columns were then removed. Subsequently, I conducted a thorough examination of the 'score' column for errors. Though there were few, I developed a function to ensure that each entry's length was 4 (x.xx), rounding up to two decimal places if greater. Missing values within the 'chapters' column were substituted with the column's average value, 27.

With the dataset now devoid of empty values, I proceeded to verify the data types of each column, rectifying any inaccuracies. For instance, the 'avg_anime_score' column was incorrectly classified as an object type rather than numerical, requiring adjustment. This process resulted in a refined dataset comprising 26,862 rows and 30 columns.

---

[1] Title (rōmaji or english)
[2] Title in english
[3] Manga media type
[4] Publishing status
[5] Number of chapters
[6] Whether it is safe for work or it is R18+
[7] MAL weighted score
[8] Number of users who favorited this anime
[9] List of genres
The above definitions were taken directly from the data's source.

Now, onto the model creation phase, I was required to make some more adjustments to the dataset. Since I did not require the 'title' or 'title_english' columns, I dropped them when creating the 'X' data. Additionally, I employed one-hot encoding on the 'X' data using pd.get_dummies(). This technique transforms categorical data into a format that enhances predictive performance. The target data was originally in floating-point type, causing some errors. To address this, I applied the round() and astype() functions to round the data to the nearest whole number and convert it to an integer type.

With these preparations completed, I proceeded to the model building phase. I utilized Python's train_test_split() function to divide the data into a training set and a test set. I prefer an 80/20 split and set the shuffle parameter to true for randomness. Given that my dataset contains mixed data types, I opted to begin with the K-Nearest Neighbor (KNN) model. This model is well-suited for such datasets. It yielded an accuracy of 75.5%, which, while acceptable, leaves room for improvement.

Seeking higher accuracy, I turned to the Decision Tree Classifier. This model, known for its implicit feature selection, achieved an accuracy of 91.3%, surpassing the KNN model. My next choice was the Random Forest Classifier, an ensemble learning method that combines multiple decision trees for improved performance. It delivered an accuracy of 93.5%, outperforming the previous model.

Finally, I tested the Support Vector Machines (SVM) model, but it yielded an accuracy of only 68.1%, lower than the KNN model. Therefore, I concluded that the Random Forest Classifier provided the highest accuracy among the models tested and concluded that it was the best model for delivery to the anime executives.

The analysis and model building process provided valuable insights into the effectiveness of different machine learning models for my dataset, ultimately guiding the selection of the most suitable model for delivery. Although the Random Forest Classifier performed well, there may still be opportunities for further improvement. Techniques such as hyperparameter tuning, feature engineering, and ensemble methods could potentially enhance the model's performance even further. A model consistently outputting accuracies higher than 93.5% would be extremely potent. I would recommend tinkering with the model and trying out these techniques. I would even try different variations of the Random Forest algorithm, such as Extremely Randomized Trees (ExtraTrees), which might yield better results.

I encountered many last-minute challenges that altered my analysis. I wish I had been able to use the exact scores from the anime dataset instead of taking the average of all scores. Many rows needed their average anime score column values to be replaced with their score values, and vice versa. Having these exact values from the start would have improved the accuracy of the model. Additionally, many values within the 'chapters' column were missing, and I wish that had been up to date as well.

What significantly impacted my analysis was the fact that it was not just a couple of rows that needed data replacement, but many, many rows did. I also had to drop hundreds of rows that would have enhanced the model building process. I lost a multitude of potentially useful data to 'NaN' values within the columns.

The last-minute conversion of the target column from floating point to integer type also affected the model creation. Although I am not entirely sure what the error was, I was forced to make that change to continue the project. It seemed a bit odd because I was able to leave the 'score' column as a floating point data type. I know that the accuracy of the model would have

changed, for better or worse, had I been able to leave the target column as is, instead of converting it into an integer data type.

I believe that having these inaccuracies corrected and missing data filled in would have provided a much higher accuracy percentage from the Random Forest Classifier, which would have definitely pleased my fictional anime executives. Nonetheless, it was a pleasure to work on a project like this from start to finish.