Navigating the Anime Universe

The Creation and Implementation of Otaku Castle's Recommender System

Task 3

Joshua C. Rogers

Western Governors University

Table of Contents

A. Project Highlights

The company Otaku Castle would like to increase interaction with visitors to its website. One way to do that is to create an anime recommender system. This system would use supervised machine learning to let visitors/customers select a few anime they have watched and give those a rating. The machine learning algorithm would then select and present several other anime the visitor might be interested in watching based on their initial selections and ratings.

The scope of the project is the creation of a machine learning model (with necessary cleaning, examination, and pre-processing of the dataset) to provide users recommendations of anime based on their preferences, available to them in an interactive webpage.

In this project the Agile method is modified to fit a single developer who is also one of the end users of the project. This iterative approach is suitable for the project as several parts, using divers technologies, must work together for the end result to work properly.

The tools used to create the solutions in this project are as follows:

1. Download of the original dataset from Kaggle: myanimelist-dataset Three files are required: anime-dataset-2023.csv, users-details-2023.csv, and users-score-2023.csv.
2. Jupyter Notebooks run within Anaconda to perform web scraping tasks and data exploration, cleaning and pre-processing:

- Pandas
- Numpy
- Matplotlib
- Beautifulsoup4
- PyTorch (with CUDA support)

3. Microsoft Visual Studio 2022 to write the necessary Python (version 3.9) scripts:
    - PyTorch to create and run the NCF model.
    - Flask to process and display the webpages that provide the model with user ratings, supply those ratings to the model, and provide recommendations to the user after the model has made its predictions.

B. Project Execution

The Project Plan:

- The goal of this project is to provide users of the website www.otakucastle.com a page where they can select anime they have seen, give those anime a rating of their own and they will be given a list of five anime they may be interested in watching next.
- The objective is to create a machine learning system that takes a dataframe of anime and the user rated scores and then is able to output several anime a user may be interested in watching next based on scores they give to a selection of anime presented to them.
- The deliverable for this project is a machine learning model and associated web pages for user input and display of results obtained by the model.

The Project Planning Methodology:

- Product Backlog Creation: Formulate a detailed list that captures all the essential features, enhancements, and project requirements. This list should be prioritized based on their criticality, interdependencies, and their potential value to the user.
- Planning: Strategically dissect the product backlog into smaller, more actionable segments. The inaugural sprint should primarily focus on laying down the foundational

infrastructure for the machine learning model and crafting an intuitive web page for user input.

- Sprint Execution: Dedicate efforts to the specified tasks, which involve rigorous data cleaning, meticulous preprocessing, the initial stages of model training, and the comprehensive design of the web page's user interface.
- Testing and Feedback: Upon culmination of each sprint, undertake a thorough evaluation of the features that have been developed. In the context of this project, this involves a critical assessment of the operational efficacy of the recommendation system and an examination of the user experience offered by the web page.
- Iterate: Subject the model and web pages to a stringent functionality test. Implement essential modifications or rectifications based on findings, subsequently updating the product backlog. Strategically plan for the subsequent sprint and reinitiate the iterative process.
- Release: Upon meticulous testing and validation of a robust set of features, roll them out to the end-users. For this project, this translates into the live deployment of the recommendation system on the website, enabling users to seamlessly engage with it
- Maintenance and Continuous Improvement: Adopt a proactive approach to constantly monitor the system's performance metrics and actively solicit user feedback. Assimilate this feedback into upcoming sprints with an aim to perpetually refine the system and introduce novel features.

Project Timeline And Milestones:

Milestone or deliverable	Duration	Projected start	Anticipated end
Deliverable: Interactive Recommender System	20 days	09/20/2023	10/10/2023
Milestone 1: Data Exploration	4 days	09/20/2023	09/23/2023
Milestone 2: Model Creation	10 days	09/24/2023	10/03/2023
Milestone 3: Webpage Creation	2 days	10/04/2023	10/05/2023
Milestone 4: Test and Deployment	4 days	10/06/2023	10/10/2023

This timeline is different from my Task 2 timeline. The original timeline was an estimate, but when it came to coding the files, creating the model, and running tests along the way, it was found to take longer than originally projected.

C. Data Collection Process

The data was selected from Kaggle: myanimelist-dataset, 3 CSV files - anime-dataset-2023.csv, users-details-2023.csv, and users-score-2023.csv. Out of the 2 similar options, the data set chosen was the most complete and up to date. This did not deviate from the original plan in Task 2 as it had already decided to use this data for the project before starting.

There were no obstacles to collecting the data: it was publically available. There were no data governance issues either. The data was already public domain and any potential personally identifiable information is in the users-details-2023.csv. This CSV (and its dataframe) were not used after data exploration. It did not contain any relevant data for the model.

C.1 Advantages and Limitations of Data Set

The data set has many advantages and few limitations. The biggest advantage is that it contains not only a very complete list of anime, but also a huge collection of ratings for most of the anime by a large number of users. Additionally, the data contains a description of each anime, an image link, genres, and a host of other information that could be displayed. While not all the data within the data set was of use for this specific project, much of the data could be used in other ways throughout the Otaku Castle website.

The biggest limitation is that this data set is fixed to the time it was collected. New anime are created consistently, and users are rating them. The data set does not provide the original data collection methods that would enable updates to the data set on a regular basis. To include new anime and additional ratings, Otaku Castle will need to provide its own method to extract any new data, incorporate it into the existing data set and retrain the model.

D. Data Extraction and Preparation

As mentioned previously, the data comes from Kaggle. It consists of three CSV files: anime-dataset-2023.csv, users-details-2023.csv, and users-score-2023.csv. The files were relatively clean, but required changes to function as the source for a machine learning model, and for future data availability. The tools used for cleaning and pre-processing are:

1) Jupyter Notebooks for Pull_Images.ipynb and AnimeRecommender.ipynb
2) Within the notebooks, Pandas, Numpy, Time and Matplotlib

The amount of data within the three datasets went far beyond the scope of this project. The datasets themselves required little cleaning, mainly centered around 'UNKNOWN' values, some of which had to be a number for the model, and some to fix naming issues. There were also a significant number of columns between the three datasets that, while useful information, was more information than needed for the recommendation system.

It was discovered during the cleaning and exploration phase that users-details-2023.csv was unnecessary. That file contained information that would have been useful had the project wanted to provide some sort of recommendations based on the country, age, or gender of the user, but there was a significant amount of missing data within these columns. However, during cleaning, I was able to confirm that all the users in both the ‘scores’ and ‘details’ datasets matched. This made eliminating the ‘details’ dataset easy as there was no need to combine any of the data into the ’scores’ dataset.

The ‘scores’ dataset was clean, but included columns not necessary for the end result of the project. In order to reduce the file size (and increase processing speed at the same time), these columns were dropped and a new CSV was created for use when running the model.

The ‘anime’ dataset includes a column called ‘Image URL’ which provides a link to an image for each anime in the dataset. This column pointed to the original location of the images, hosted on the My Anime List server. In order to make the images available on the Otaku Castle website (eliminating a lookup on a website outside the control of Otaku Castle) a webscraping script (Pull_Images.ipynb) was created and executed. The script was only run once, at the beginning of the project. This script:

1) Creates a directory to store the images.
2) Downloads each image (at 3 second intervals).
3) Renames the image to the ‘English name’ title of the anime.

4) Updates the ‘Image URL’ column with new site and image name information.
5) Saves the updated dataset to a new CSV for use with the remainder of the project.

After this step is complete, the remainder of the cleaning, exploration and visualization is performed within AnimeRecommender.ipynb and a new CSV is written for use by the model.

The AnimeRecommender notebook performs several functions:

1) Imports the 3 CSV files
2) Examines the structure of each CSV (as a dataframe)
3) Cleans each dataframe, then saves the updated file to CSV
4) Performs data exploration on the remaining 2 dataframes, including visualizations
5) Performs pre-processing and encoding in preparation for the model
6) Saves the cleaned and pre-processed files as CSV for use with the model.

E. Data Analysis Process

E.1 Data Analysis Methods

Analytical Method: Neural Collaborative Filtering (NCF)

The primary analytical method employed was the Neural Collaborative Filtering (NCF) approach, designed for the implementation of the anime recommender system. This method was chosen to support the hypothesis because it is possible to enhance user engagement by not only bolstering user interaction but also personalizing the user experience on Otaku Castle. The NCF model was deemed appropriate for several reasons:

1. Complex Pattern Recognition: NCF can capture intricate patterns in user-item interactions, going beyond the capabilities of many other recommendation system algorithms.
2. Scalability: The NCF model can efficiently scale, absorbing new data without a marked degradation in performance, ensuring that as Otaku Castle expands, the recommender system remains robust.
3. Integration & Flexibility: NCF offers the flexibility to be integrated with other neural network architectures. Additionally, as Otaku Castle grows, features like purchasing data can be incorporated into the model, enhancing the recommendation quality.

To support the NCF method, various tools and environments were utilized:

1. Data source: Kaggle.com; myanimelist dataset (linked above)
2. Data Processing & Exploration: Jupyter Notebooks, run within the Anaconda environment, were instrumental in web scraping, data exploration, cleaning, and preprocessing. The tools leveraged included Pandas, Numpy, Matplotlib, Beautifulsoup4, and PyTorch.
3. Model Implementation & Deployment: Microsoft Visual Studio 2022 was used to script the solutions in Python (version 3.9). PyTorch was pivotal in creating and running the NCF model, while Flask was used to display web pages that interface with the model, providing user ratings and subsequent recommendations. As a side note, running the model without CUDA (Nvidia GPU support) would have taken significantly longer.

Evaluation Methods & Metrics:

1. Model Description: The model is classified under Supervised Learning, specifically Regression. It predicts continuous rating values based on user-anime interactions, using the Neural Collaborative Filtering (NCF) algorithm.
2. Metrics: The primary metric is Loss, which quantifies the disparity between the model's predictions and actual user ratings. The objective is to achieve a training loss near 1.6, ensuring the validation loss mirrors the training values.
3. Justification: The project's primary goal is to formulate an anime recommendation system. Given that the task is to predict continuous values (ratings), it naturally falls under regression. NCF, tailored for recommendation systems, offers the combined benefits of matrix factorization and deep learning, ensuring precise and personalized recommendations. The loss metric directly reflects the alignment between predicted and actual ratings, crucial for enhancing user experience on Otaku Castle.

E.2 Advantages and Limitations of Tools and Techniques

1) Kaggle Dataset:

Advantage: Rich Data Source. Kaggle is a renowned platform for data science competitions and datasets. The datasets available, like the myanimelist-dataset, often come from genuine sources, ensuring authenticity and richness in data content.

Limitation: Static Nature. Once downloaded, the dataset is static and might not reflect the most recent user-anime interactions or ratings, potentially impacting the model's current relevance.

2) Jupyter Notebooks (within Anaconda):

Advantage: Interactive Environment. Jupyter Notebooks provide an interactive environment, allowing for real-time code execution, visualization, and documentation, all in one place.

Limitation: Performance. For extremely large datasets or intense computations, Jupyter Notebooks might experience performance issues or memory constraints.

Tools within Jupyter Notebooks:

Pandas & Numpy:

Advantage: Data Manipulation. These libraries provide robust tools for data manipulation, analysis, and mathematical operations, streamlining the data preparation process.

Limitation: Memory Consumption. While efficient, they can be memory-intensive for very large datasets, potentially slowing down operations.

Matplotlib:

Advantage: Visualizations. Offers a wide array of plotting tools for data visualization, aiding in data exploration and insights.

Limitation: Styling. While versatile, it might require more effort for advanced styling or interactive plots compared to other visualization libraries.

Beautifulsoup4:

Advantage: Web Scraping. Enables efficient extraction of data from web pages, expanding the data sources beyond just the initial dataset.

Limitation: Website Changes. If the structure of a website changes, the scraping code might break, necessitating regular maintenance.

PyTorch:

Advantage: Deep Learning & GPU Acceleration. PyTorch is a dynamic deep learning framework that supports GPU acceleration, allowing for faster model training and inference.

Limitation: Learning Curve. For those accustomed to static computational graphs (like in TensorFlow), PyTorch might present an initial learning curve.

3) Microsoft Visual Studio 2022:

Advantage: Integrated Development Environment (IDE). Offers a comprehensive environment for writing, debugging, and deploying code, enhancing developer efficiency.

Limitation: Resource Intensive. Being a full-fledged IDE, it can be more resource-intensive compared to lightweight code editors.

Tools within Microsoft Visual Studio 2022:

Flask:

Advantage: Lightweight & Flexible. Flask is a micro web framework in Python, allowing for quick setup and deployment of web applications.

Limitation: Scalability. For large-scale web applications with multiple functionalities (like this could become in the future), a more comprehensive framework like Django might be better suited.

Python:

Advantage: Versatility & Libraries. Python is renowned for its versatility, catering to a wide array of tasks from web development to machine learning. Its vast ecosystem of libraries, such as Pandas, Numpy, and PyTorch, facilitates swift development and data analysis.

Limitation: Speed. While Python is excellent for development speed and versatility, it is not as fast as languages like C++ or Java in terms of execution. This can be a limitation for real-time applications or highly computational tasks.

E.3 Application of Analytical Methods

Method: Neural Collaborative Filtering (NCF) for Anime Recommender System

Steps used:

1. Dataset Acquisition: The original dataset, "myanimelist-dataset", was sourced from Kaggle. This dataset contains user-anime interactions and ratings.
2. Data Processing: Using Jupyter Notebooks within the Anaconda environment, various tasks were performed:
   - Web Scraping: Data was updated through web scraping tasks.
   - Data Cleaning: Any inconsistencies, missing values, or outliers in the data were addressed.
   - Data Exploration: The dataset's structure, missing values, and potential patterns were explored.
   - Data Pre-processing: The data was prepared for model training, which may involve normalization, encoding, or splitting the dataset into training and validation sets.
3. Model Implementation: Using Microsoft Visual Studio 2022, the following was accomplished:

- NCF Model Creation: The NCF model was designed and implemented using PyTorch with CUDA support. This ensures that the model can take advantage of GPU acceleration for faster training.
- Model Training: The NCF model was trained on the prepared dataset, aiming for a loss value that quantifies the difference between predicted and actual user ratings.
- Deployment: The trained model was integrated into a Flask web application. This web app processes user ratings, feeds them to the NCF model, and provides anime recommendations based on the model's predictions.

Requirements for the method:

1. Authentic Dataset: The integrity and authenticity of the data from Kaggle is crucial. It must reflect user-anime interactions and ratings.
2. Appropriate Data Processing Tools: Libraries like Pandas, Numpy, Matplotlib, and Beautifulsoup4 are essential for various data processing tasks.
3. Deep Learning Framework: Given the choice of the NCF model, a deep learning framework like PyTorch is necessary for model creation, training, and inference.
4. Web Framework for Deployment: To bring the solution to end-users, a web framework like Flask is required to deploy the model and facilitate interactions with users.

Verification:

1. Dataset Authenticity: The source of the dataset, Kaggle, is renowned for hosting authentic datasets, ensuring its credibility.
2. Data Processing Verification: The effectiveness of data processing was verified through exploratory data analysis (using Matplotlib) and initial model performance metrics.
3. Model Performance: The NCF model's performance was gauged using the loss metric. Achieving a training loss near 1.6 and ensuring that validation loss stays close proves the model's efficacy.
4. Deployment Verification: Successful integration of the model within the Flask application and its ability to provide recommendations based on user input verifies the deployment step.

F Data Analysis Results

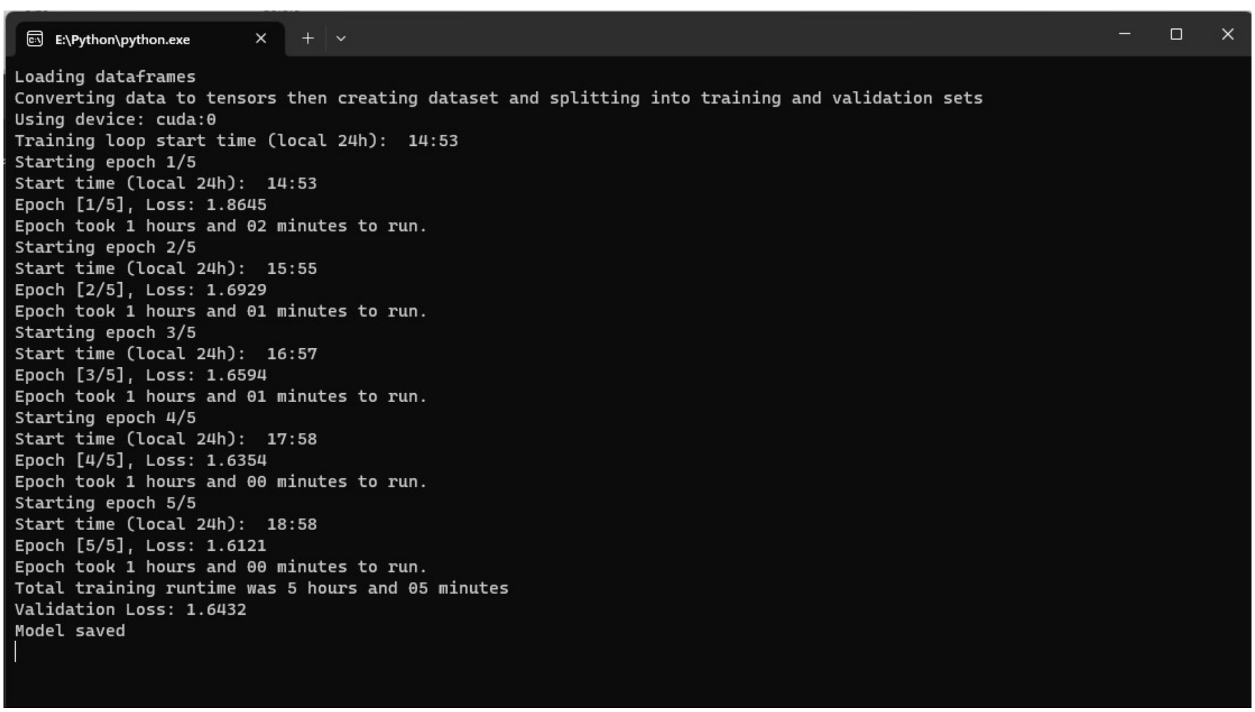
F.1 Statistical Significance

Type of Model: The model employed is a Supervised Learning - Regression. Its purpose is to predict continuous rating values derived from user-anime interactions.

Algorithm and Process: The primary algorithm used for this model is Neural Collaborative Filtering (NCF). NCF is a modern approach that integrates matrix factorization and deep learning techniques, making it particularly suitable for recommendation systems. By leveraging both these methodologies, the model is adept at recognizing both overt patterns in user-item matrices and subtle trends through deep learning processes.

Metric Used to Assess Performance: The performance of the model is evaluated using the metric of Loss. This metric quantifies the difference between the predicted ratings and the actual user ratings. The smaller the loss, the closer the model's predictions are to the actual values, indicating a higher level of accuracy.

Benchmark for Success: The benchmark set for the performance of this model is a training loss near 1.6. Alongside this, it is vital to ensure that the validation loss is in proximity to the training loss values. This approach guarantees that the model is not only accurate with the data it has been trained on but is also reliable when predicting ratings based on new, previously unseen user-anime interactions. The image below shows the training results and the evaluation result.



E:\Python\python.exe

```
Loading dataframes
Converting data to tensors then creating dataset and splitting into training and validation sets
Using device: cuda:0
Training loop start time (local 24h):  14:53
Starting epoch 1/5
Start time (local 24h):  14:53
Epoch [1/5], Loss: 1.8645
Epoch took 1 hours and 02 minutes to run.
Starting epoch 2/5
Start time (local 24h):  15:55
Epoch [2/5], Loss: 1.6929
Epoch took 1 hours and 01 minutes to run.
Starting epoch 3/5
Start time (local 24h):  16:57
Epoch [3/5], Loss: 1.6594
Epoch took 1 hours and 01 minutes to run.
Starting epoch 4/5
Start time (local 24h):  17:58
Epoch [4/5], Loss: 1.6354
Epoch took 1 hours and 00 minutes to run.
Starting epoch 5/5
Start time (local 24h):  18:58
Epoch [5/5], Loss: 1.6121
Epoch took 1 hours and 00 minutes to run.
Total training runtime was 5 hours and 05 minutes
Validation Loss: 1.6432
Model saved
```

Conclusion: Given the set benchmarks and the performance metrics used, we can conclude that the model's training loss is close to the targeted 1.6 and that the validation loss remains consistent with the training loss, showing the model is successfully making accurate predictions. This would support the hypothesis that the Neural Collaborative Filtering approach, when combined with a well-defined loss metric, can effectively predict user ratings based on their interactions with anime titles. The end goal is to enhance the user experience on Otaku Castle by providing them with precise and tailored anime recommendations, and achieving these benchmarks validates our model's capacity to do so.

F.2 Practical Significance

In the sprawling universe of anime, potential viewers often grapple with the sheer abundance of choices. Otaku Castle's vision of heightening website interactions is materialized through the inception of an anime recommender system. Harnessing the power of supervised machine learning, the system illuminates a path for visitors, guiding them through a curated journey based on their unique tastes and preferences.

Criteria for Assessing Practical Significance:

- User Engagement: A tangible uptick in user interaction on the website post-implementation of the recommender system. Enhanced user engagement is an indicator of the system's efficacy in tailoring recommendations that resonate with individual preferences.

- User Retention: Monitoring the return rate of users can offer insights into the recommender system's success. A higher retention rate would be indicative of users finding value in the personalized recommendations, fostering loyalty to Otaku Castle.
- Feedback and Reviews: User testimonials and feedback can serve as qualitative measures of the system's impact. Positive reviews centered around the accuracy and relevance of the recommendations can affirm the practical significance of the model.
- Conversion Rates: Monitoring any potential increase in purchases or other desired user actions directly resulting from the recommendations. This would be a direct measure of the system's impact on user behavior.

Much of the criteria above requires measurements through data analytics. To fully validate the results of the model across a wide swath of users will take months of site integration, user feedback, and website analytics. During this phase, a select group of users will have access to the model and its associated pages to provide not only model result feedback but also aesthetics.

Otaku Castle's aspiration to bolster website interaction is linked to its desire to provide users with a seamless and tailored experience. The recommender system, based on Neural Collaborative Filtering, stands as a testament to this vision. By allowing users to rate a select set of anime, the system, in turn, curates a list of recommendations that echo their unique tastes. The overarching research question centers around the potential of machine learning to enhance user engagement. The implemented system directly addresses this, aiming to transform Otaku Castle into a revered sanctuary for anime aficionados.

Imagine a user, Cera, a fan of the fantasy genre. On her first visit to Otaku Castle, she rates a few of her favorite fantasy anime. The recommender system, harnessing the power of NCF, curates a list of lesser-known fantasy anime that aligns with her tastes. Intrigued, Cera delves deeper, spending more time on the site. On her subsequent visits, she not only rates more anime but also explores merchandise related to her recommendations. The system's precision in pinpointing her preferences ensures her loyalty to Otaku Castle, and she soon becomes a regular visitor and buyer, trusting the platform for her anime recommendations and merchandise.

F.3 Overall Success

The success of the Otaku Castle project, aimed at creating an effective and tailored anime recommender system, can be evaluated through both statistical and practical lenses as observed from sections F1 and F2.

Statistical Success (F1):

Achievement of Model Criteria: A working Neural Collaborative Filtering (NCF) model was developed utilizing the cleaned and processed Kaggle dataset. NCF, being an integration of matrix factorization and deep learning, made the model particularly adept at understanding explicit and subtle trends in user-anime interactions.

Performance Metrics: The key metric used to gauge the performance was Loss. The model was designed to minimize the difference between predicted and actual user ratings.

Benchmark Realization: The set benchmark for the model was a training loss nearing 1.6. The results showcase that the model's training loss approached this target, and equally importantly, the validation loss stayed consistent with the training loss. This consistency indicates that the model's predictions are not just accurate for the training set, but also reliable when predicting unseen user-anime interactions.

Practical Significance (F2):

User Experience and Engagement: The vision of Otaku Castle to heighten user interactions saw its realization with the recommender system's deployment. By allowing users to rate select anime, the system, in response, offers a curated list of recommendations that mirrors individual tastes. This is a direct endeavor to make the platform more user-centric and interactive.

User Retention and Feedback: While the full-fledged effects on user retention will be evident over a more extended period, the recommender system is anticipated to boost user return rates. Feedback and testimonials from users, particularly on the accuracy and relevance of recommendations, will further validate the model's practical significance.

Conversion Rates and Business Impact: By monitoring any potential spikes in purchases or desired user actions resulting from the model's recommendations, Otaku Castle can directly assess the system's influence on user behavior.

Personalized User Stories: Consider a user like Cera. The precision of the recommender system in recognizing her fantasy genre inclination and suggesting lesser-known titles aligned with her preferences exemplifies the model's effectiveness. Such tailored experiences ensure increased user loyalty, more extended site interactions, and potentially higher conversion rates.

Future Insights and Feedback Collection: The post-launch phase is dedicated to collecting user feedback and website analytics over several months. This continuous improvement cycle will further refine the system and its offerings.

In summary, the Otaku Castle project was undeniably successful from both statistical and practical viewpoints. The effective deployment of the Neural Collaborative Filtering model, combined with the designed user interactivity elements, reaffirms the project's objective to transform Otaku Castle into a premier destination for anime enthusiasts.

G. Conclusion

G.1 Summary of Conclusions

The project successfully implemented a Supervised Learning - Regression model utilizing the Neural Collaborative Filtering (NCF) approach. This model was adept at predicting user ratings based on user-anime interactions, leveraging both matrix factorization and deep learning techniques.

The primary metric, Loss, was vital in gauging the model's accuracy by quantifying the difference between predicted and actual user ratings. The achieved training loss neared the set benchmark of 1.6, indicating the model's robust accuracy during training. Additionally, the consistent validation loss showcased the model's ability to generalize its predictions, even for unseen data.

The recommendation system provides a tailored path for users by curating anime suggestions based on their unique tastes and preferences, which will ideally result in a tangible uptick in user engagement. The project envisions users finding significant value in the system, with positive feedback and heightened retention rates.

The project not only established a backend model but also incorporated user-friendly interfaces with ratings.html and watchnext.html. While their current appearance is simplistic, the eventual integration into the Otaku Castle website will provide a seamless user experience, enriched by site-wide CSS stylings and familiar layouts.

With an extended integration phase, the project aims to gather comprehensive data analytics, user feedback, and more to refine the recommendation system. This continuous cycle of improvement will further boost user engagement, loyalty, and business metrics.

In summary, the Otaku Castle project adeptly marries statistical robustness with practical utility. By leveraging modern machine learning techniques and user-centric designs, the initiative is poised to dive into the anime recommendation landscape, promising users curated suggestions and an enriched browsing experience.

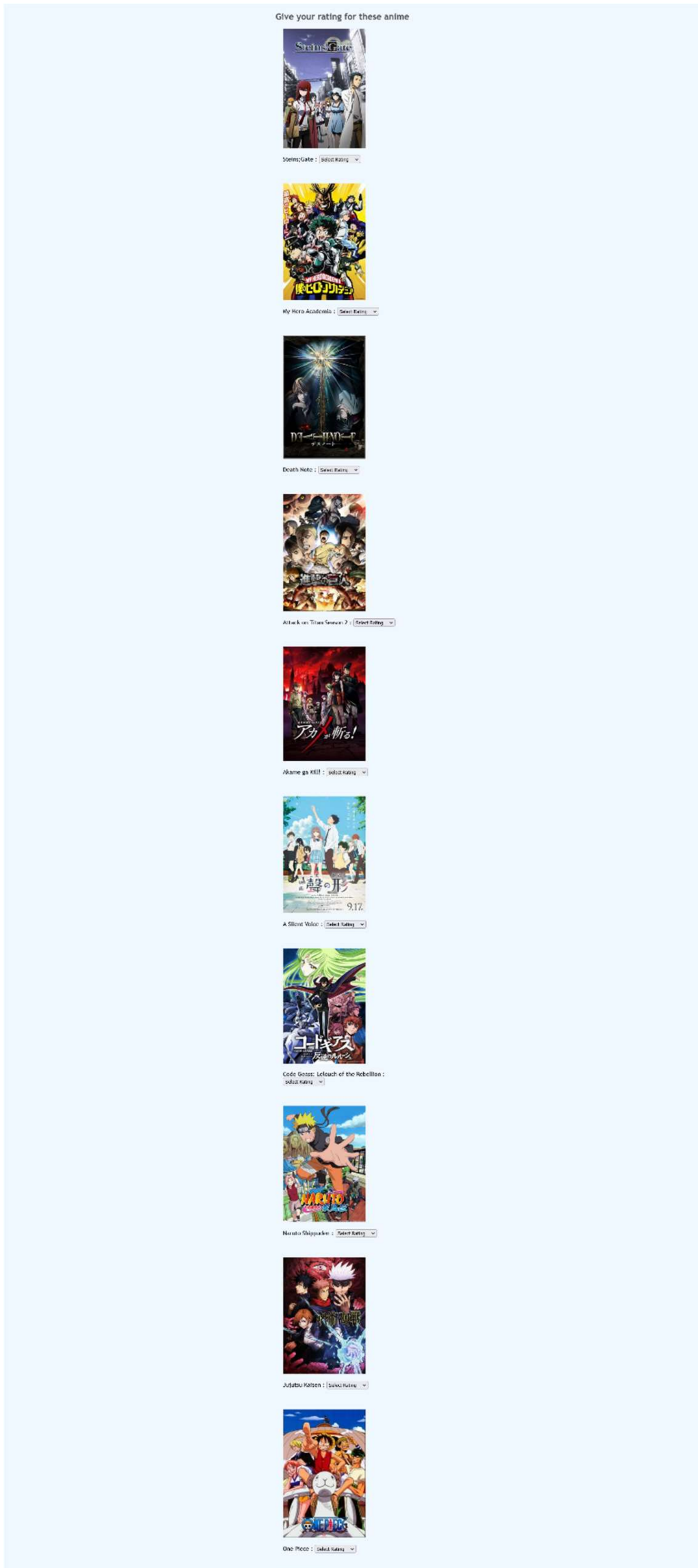
G.2 Effective Storytelling

Below are snapshots from our active model interface, captured using Firefox's built-in screenshot tool.

For the backend framework, Flask was used to display two main pages: ‘ratings.html’ and ‘watchnext.html’. These pages were developed in Microsoft Visual Studio 2022, using simple HTML paired with a temporary CSS for basic styling—font and basic layout.

The initial page, ratings.html, serves users with 10 random anime picks from the model's top 30. Users are prompted to rate these selections on a scale of 1-10 or opt for a 'Haven't seen' choice. After rating, a 'Get Recommendations' button lets users get results based on their inputs.

While these screenshots provide a glimpse, their intricate details are better showcased in the upcoming video presentation for clarity.


Give your rating for these anime
Steins;Gate : Select Rating
My Hero Academia : Select Rating
Death Note : Select Rating
Attack on Titan Season 2 : Select Rating
Akame ga Kill! : Select Rating
A Silent Voice : Select Rating
Code Geass: Lelouch of the Rebellion : Select Rating
Naruto Shippuden : Select Rating
Jujutsu Kaisen : Select Rating
One Piece : Select Rating

After users have rated the anime, Flask channels the data and navigates to the watchnext.html page. This page presents 5 anime recommendations tailored to the user's ratings. Additionally, it offers a brief description from the clean_anime dataframe, detailing the genres and specifying whether the recommendation is a TV/Series, Movie, OVA, or other formats.

While these pages currently sport a minimalist design, once integrated into the Otaku Castle website, they'll nestle within the site's structure—complete with headers, footers, and menus. The established site-wide CSS will naturally extend, bestowing the familiar fonts, image positioning, text styling, and overall aesthetic consistent with Otaku Castle's theme.

Your Anime Recommendations

Rurouni Kenshin

Genres: Action, Adventure, Comedy, Romance
Type: TV

Description: In the final years of the Bakumatsu era lived a legendary assassin known as Hitokiri Battousai. Feared as a merciless killer, he was unmatched throughout the country, but mysteriously disappeared at the peak of the Japanese Revolution. It has been ten peaceful years since then, but the very mention of Battousai still strikes terror into the hearts of war veterans. Unbeknownst to them, Battousai has abandoned his bloodstained lifestyle in an effort to repent for his sins, now living as Kenshin Himura, a wandering swordsman with a cheerful attitude and a strong will. Vowing never to kill again, Kenshin dedicates himself to protecting the weak. One day, he stumbles across Kaoru Kamiya at her kendo dojo, which is being threatened by an impostor claiming to be Battousai. After receiving help from Kenshin, Kaoru allows him to stay at the dojo, and so the former assassin temporarily ceases his travels. Rurouni Kenshin: Meiji Kenkaku Romantan tells the story of Kenshin as he strives to save those in need of saving. However, as enemies from both past and present begin to emerge, will the reformed killer be able to uphold his new ideals?

Gungrave

Genres: Action, Drama, Sci-Fi
Type: TV

Description: Brandon Heat and Harry MacDowel, two friends so close they could be called brothers, receive an abrupt and violent reminder one fateful day of how appallingly merciless the world around them can be. Their whole lives before then were simple and easygoing, consisting largely of local brawls, seducing women, and committing petty theft to make a living and pass the time. What they failed to realize is that in this cruel world, happiness is fleeting, and change is inevitable. Enter Millennion, the largest and most infamous mafia syndicate in the area, which accepts Brandon and Harry into their ranks and starts them at the bottom of the food chain. Harry has ambitions to ascend the ranks and one day replace Big Daddy as the supreme leader of Millennion, while Brandon only wishes to support his friend and appease Big Daddy who has taken custody of the woman Brandon loves. Based off the third-person shooter video game under the same name, Gungrave is an epic story of friendship, betrayal, and avarice that spans the course of several years, ultimately tying back to the gripping and foreboding first episode, all the while building up to the story's thrilling conclusion.

Bottle Fairy

Genres: Comedy, Fantasy
Type: TV

Description: Set in the year 2004, Binzume Yousei is a slice-of-life fairy tale that revolves around four fairies, each represented by four unique colors as seen with their magical bottle jars. These fairies are the extremely peppy Kururu, the reserved and feminine Chiriri, the samurai-loving tomboy Sarara, and the quiet yet quirky Hororo. Fascinated by the human world, these fairies arrived from the fairy world in hopes of participating in the annual traditions and overall way of human life. However, they have a very limited understanding of the human world. Luckily, they are befriended and guided by two humans—"Sensei-san," a university student who they live with, and a first-grade girl they call "Tama-chan," who is sometimes as naive as the fairies themselves. Though these bottle fairies have strange ideas and sometimes have difficulty understanding this new world, they try to make the most of the human experience in their own cute little ways.

Noein: to your other self

Genres: Adventure, Drama, Sci-Fi
Type: TV

Description: During their last summer of elementary school, four friends decide to undertake a test of courage at their local graveyard. Before the test begins, Haruka Kaminogi makes a last effort to pull Yuu Gotou away from his controlling mother. While doing so, Haruka suddenly has a strange vision of blue snow followed by the appearance of an imposing silver-haired man. Later, a similar vision occurs at the graveyard to both Haruka and her friends before they try to escape what they assume are ghosts. Unbeknownst to the children, the people who appeared before them are Dragon Soldiers: an elite military group from a dimension known as La'cryma. The soldiers have traveled to this dimension to secure the "Dragon Torque"—an entity they believe to be their last hope for survival. However, both the Dragon Soldiers and Haruka are shocked to learn that the Dragon Torque is Haruka herself. She attempts to escape from the Dragon Soldiers as she finds her own last ray of hope—the strange silver-haired man who claims to be another version of Yuu himself.

Negima!? Haru Special!?

Genres: Comedy, Fantasy, Romance, Ecchi
Type: OVA

Description: Sometime after all the events which took place during Negi's first year as a teacher, Class 2-A goes on a school trip to a beautiful island for Spring break. Everything is fine now, but troubles continue for Negi, because he's still the main atraction of the class. But he has something else to worry about, Asuna is angry at him because he told her she had nothing to do with his studies as a wizard. During the trip, the situation gets intense as Asuna's friends try to help Negi make up with her. (Source: ANN)

G.3 Recommended Courses of Action

Recommendation 1: User-Driven Refinement and Personalization

Description: Implement a feedback loop where users can provide direct feedback on the accuracy and relevance of the anime recommendations they receive. This would involve a simple "Did you find these recommendations relevant?" survey or rating system for each recommendation.

Relation to Research Question: The primary aim of the Otaku Castle project is to enhance interaction with its website visitors. By introducing a user-driven feedback mechanism, visitors become more engaged, not just passively receiving recommendations but actively participating in refining them. This continuous interaction would serve the dual purpose of optimizing the recommendation system and increasing user engagement, directly addressing the core research question.

Recommendation 2: Diversified Content Integration

Description: Apart from just anime recommendations, integrate related content recommendations such as manga adaptations, merchandise related to the anime, or forums/discussions centered around the recommended titles. This holistic approach offers a richer and more immersive experience for the visitors.

Relation to Research Question: The essence of the research question revolves around augmenting interaction. By presenting diversified content connected to the recommended anime, visitors are enticed to explore more than just the recommended titles. They might delve into related manga, engage in discussions, or even make purchases, significantly increasing their interaction duration and depth on the Otaku Castle website.

H Panopto Presentation

The Panopto presentation can be found at:

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=cffcd14e-e166-4d1e-8016-b09f0183e3a2#