Ultimate Fighter Championship Recommendation Engine

I.Comparative Study on Recommendation Engines without Implicit or Explicit Feedback

Alexadriana Thammavongsa, David Rush, and Abasiama Akpan,⋆

Masters of Computer Science, Northeastern University

Final Submission April 15, 2020

ABSTRACT

Aims. Compare various implementations of recommendation engines to determine which may prove most beneficial in addressing a cold start problem with a dataset without implicit or explicit feedback.
Methods. Dimensionality reduction via PCA, k-Means, k-Nearest Neighbors, and maxtrix factorization through Single Variable Decomposition and Alternating Least Squares
Results. ALS proves to be most efficient at combining concepts derived from kNN and SVD without the need for explicit feedback. kNN accurately describes relationship between nearest neighbors, and SVD requires explicit feedback created by initializing users.

Key words. UFC – MMA – Machine Learning – Recommendation Engine – Collaborative Filtering, Content-Based Filtering

1. Introduction

The Ultimate Fighter Championship (UFC) popularized Mixed Martial Arts (MMA) in 1993 with the intent to identify the most effective martial art in a fighting contest compromised of limiting rules and regulations that ensure fighter safety. Since then, the UFC has adopted twelve weight divisions, and fighters began to adopt kickboxing, Muay Thai, boxing, Brazillian Jiu-Jitsu, sambo, wrestling, judo, etc, that they deem most effective to their fighting style. Some particular fighters, such as Khabib Nurmagomedov or Conor Mcgregor, have become known for their particular fight style. Nurmagomedov, for example, is known for his dominant style of wrestling and Mcgregor his overwhelming striking capabilities.

Most veteran viewers take a liking to a specific type of fighting, but it becomes difficult for casual or new viewers to know the intricacies of each fighter's fighting style. This project aims to compare different implementations of recommendation system that produces accurate recommendations based on these styles and weight class with limited feedback.

This project only takes into consideration raw fighter statistics which greatly impacts the results produced. Users will be recommended similar fighters based on fighter's effectiveness in the cage not their social media presence. Other important factors to consider is that this data does not have ratings, recommendations, user-input already established. This project explores how different methods of recommendation systems filter data without explicit or implicit feedback and garner whether or not implementing such an action is feasible. The three main methods implemented for comparison are: k-Nearested Neighbors, Single Variable Decomposition, and Alternating-Least-Squares. Other methods implemented help visualize the data.

⋆ Just to show the usage of the elements in the author field

2. Understanding the Data

The data used from this project details two fighters performance for a specific match, and their attempted attacks and strikes have also been recorded. This data list which event it is, who was the referee, the weight class those fighters fought at, attempted and landed strikes, takedowns, and submissions, and number of wins via split decision, unanimous decision, technical knockout, knockout, or submission.

The weight class section was not listed as numerical – to which this project used to add a separate column to distinguish between gender. For example, men's and women's flyweight lists between 48 kilograms to 51 kilograms, but the data set provided by Kaggle distinguishes between the men's and women's division in a string format. The gender column was added as a boolean column, listing men as "1" and women as "0". In terms of creating a recommendation system, it is important to distinguish between the genders because the two genders may fight differently. For example, women's MMA may strike with a higher volume than men's or vise versa. If this is the case, then the recommendations for a participant in the mens' heavyweight division, where individual strikes tend to deal more damage, would likely recommend other people within that division. A male in the flyweight division may numerically resemble those in the women's division due to sheer volume.

Note that the recommendation system may not filter by gender nor is it intended to. This dataset determines similarity based on a fighter's accuracy both offensively and defensively along with their ability to successfully win. There is no reliable measure to determine a fighter's actually implemented fighting style since that requires visual input. Therefore, a fighter who successfully takes down their opponent with a 70% accuracy and wins via technical knockout is numerically similar to another fighter with those same percentages regardless of gender. In terms of visualizing the data, however, gender is taken into account.

3. Methods

A. Dimensionality Reduction - PCA
We apply dimensionality reduction to obtain an ordered list of components that account for the largest variance in the data set in order to ultimately group similar fighters based on their fighting styles. Some of the matches feature fighters with missing stance information or missing referee information. Since this is a relatively small number of matches out of the whole dataset, these matches have been dropped from the dataset. For any missing numerical data, those columns were filled in with the median value for that particular column. The purpose of the PCA plot is to map the correlation amongst the columns into a 2-D graph, and we determine the number of principal components using the scree plot and further improved by a scatterplot analysis.The scatter plot from the PCA analysis doesn't give us a full picture of the variance within the principal components but it does aid in our K-means clustering technique as described below

B. K-Means Clustering
K-means clustering has been implemented to visualize the similarity amongst the principal components. In order to chose an optimal k, we use the elbow point in the inertia graph. By using the result from the PCA into the K-means analysis, the clusters formed will provide more insights into our dataset.

C. Linear Regression
In order to visual the data further, it was important to understand the relationship between all of the different statistics to see if there is a particular feature that is most significant overall amongst fighters and whether or not a recommendation system should be centered around a particular feature. The dataset was split into two groups, based on fighter corner. In particular, this portion focused on which feature is most strong associated with winning. Furthermore, this project created categorical plots that analyzed wins, weight, and gender.

D. k-Nearest Neighbors
This method determines the k-Nearest Neighbors (kNN) in order to determine similar fighters based on their fighting statistics provided by the dataset. In this case, the attributes include both overall volume landed and attempted strikes. This allows for a larger scale understanding of how aggressive a particular fighter can be and their accuracy. Per the nature of kNN, we did not implement a training phase and saw how accurately it worked against a large dataset.

E. Matrix Factorization via Singular Value Decomposition
With our dataset represented as matrix, we can perform singular value decomposition (SVD) which allows us to decompose our dataset into component matrices of the form:
$A = U \epsilon V^T$
From this we can approximate and make predictions based on keeping the most important features in the original dataset.

F. Alternating Least Squares
The alternating least squares is another variation of matrix decomposition, similar to SVD. This can also be used to create a recommendation system but in an implicit feedback fashion. Furthermore, it can be tested on one particular feature.

4. Results

PCA

The components correspond to combinations of the original features and below are the results on visualizing the variation of the original features based on the component reduction along with the corresponding scree plot.

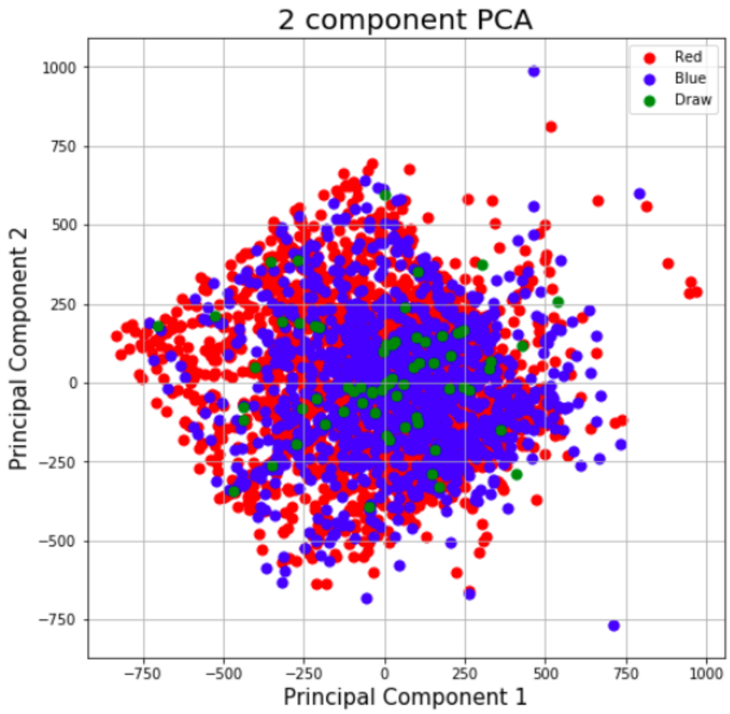

Fig. 1. Scatter plot for PCA

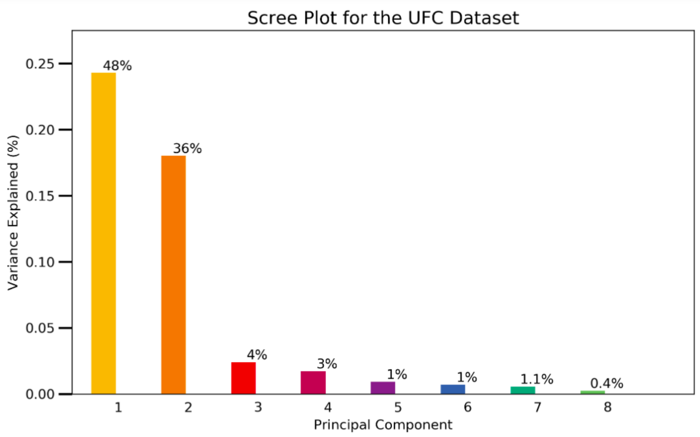

Fig. 2. Scree Plot for UFC Dataset

K-Means Clustering
Using inertia, the principal components have been fitted to the k-means algorithm to determine the number of clusters. Below also showcases the distinct clusters.

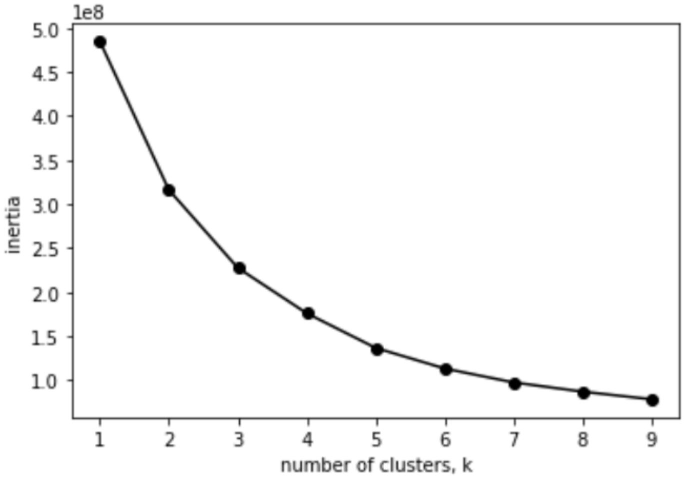

Fig. 3. Inertia plot for K-means cluster with elbow point at k=6

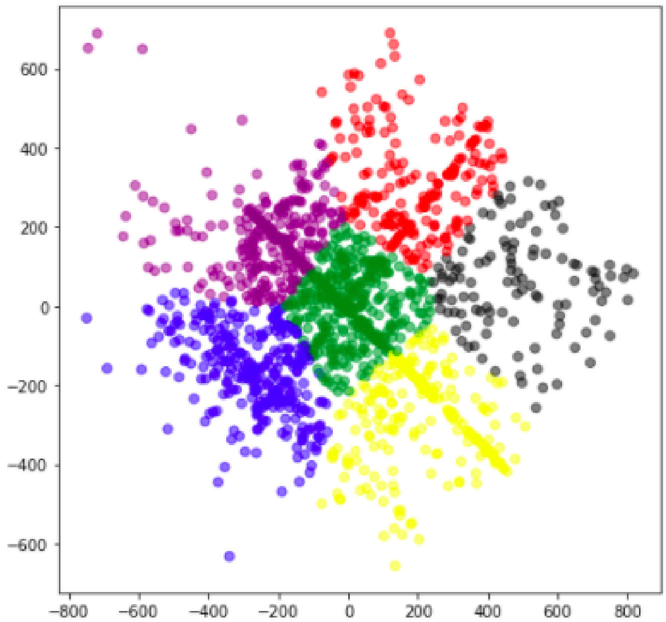

Fig. 4. K-means cluster for UFC dataset. There are 6 distinct clusters

Linear Regression

Below demonstrates two sample linear regression plots in order to determine if there is a pair of features that would be most pertinent towards either an red or blue fighter success. Below are a sample of crucial features against their ability to win.This dataset does not provide a distinction between the two. The metrics are: Coefficients, Mean Squared Error (MSE), Coefficient of Determination (Determination) and Variance Score.

For the Red Fighters:

Coefficients	MSE	Determination	Variance Score
0.24839598	1.76	.89	.89

For the Blue Fighters:

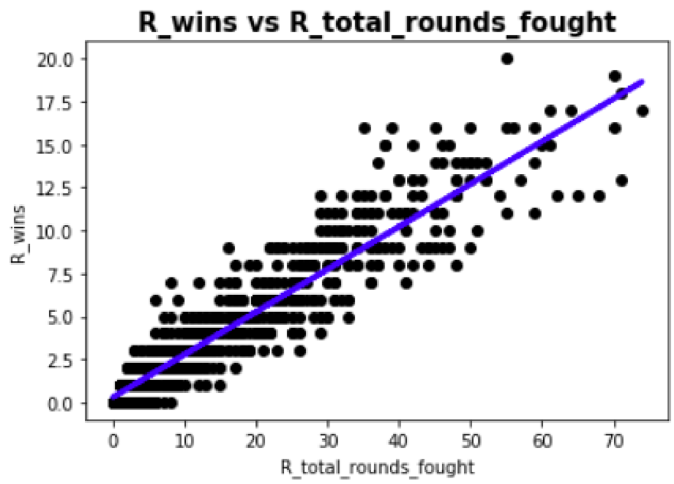

Fig. 5. Linear Regression Plot on Red Wins

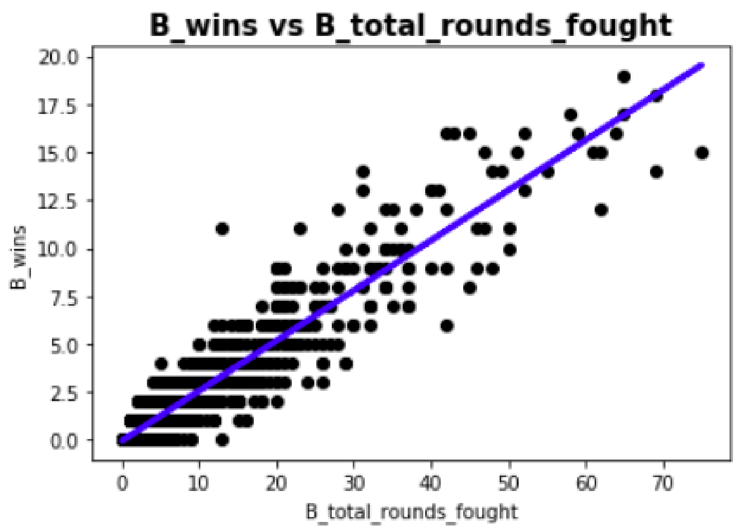

Fig. 6. Linear Regression Plot on Blue Wins

Coefficients	MSE	Determination	Variance Score
0.26128101	1.36	0.88	0.88

Average of pair-wise features:

Coefficients	MSE	Determination	Variance Score
0.44694159	13.972	0.86428	0.0898

kNN Accuracy for the blue corner:

```
KNN 0.9579579579579579
```

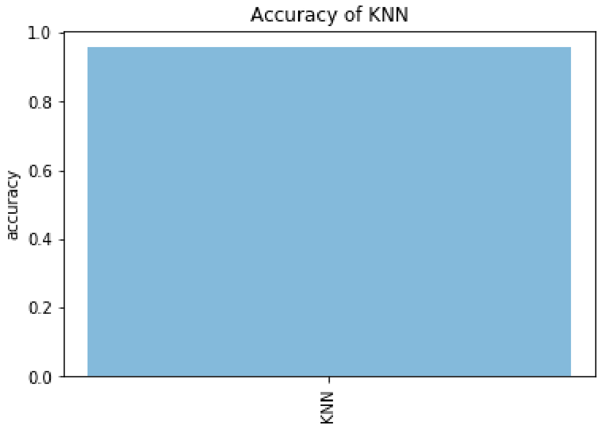

Fig. 7. Blue Corner kNN Accuracy

Accuracy for the red corner:

```
KNN 0.9369369369369369
```

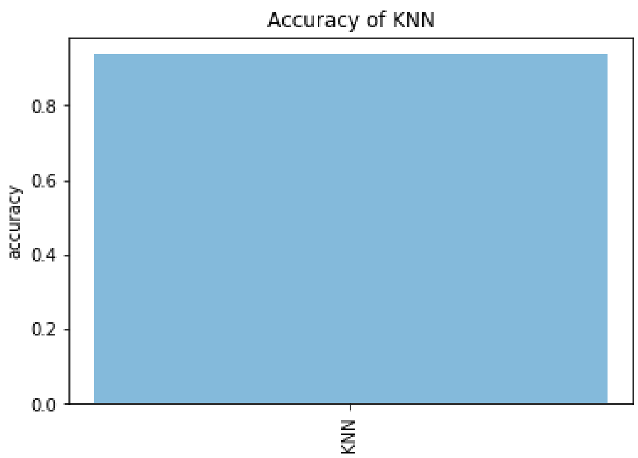

Fig. 8. Red Corner kNN Accuracy

kNN Results for Valentina Schevchenko for male recommendations:

	R_fighter	B_fighter
419	Luke Rockhold	David Branch
579	Luke Rockhold	Michael Bisping
1092	Brian Foster	Forrest Petz

Fig. 9. Schevchenko Male Recommendations

kNN Results for Valentina Schevchenko for female recommendations:

	R_fighter	B_fighter
1635	Amanda Nunes	Ronda Rousey

Fig. 10. Schevchenko Female Recommendations

Results for SVD:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	1	1	1	1	1	0	0	0	0										
2						1				1	1	1	0	0					
3															1	1	1	0	0
4		1				1													

Fig. 11. Matrix for SVD

If a figher is like, that fighter received a 1. Otherwise, 0.
The rows represent users 1-4.
The columns represent different fighters.
The key is as follows for the fighters:

1. Khabib Nurmagomedov
2. Amanda Nunes
3. Weili Zhang
4. Kamaru Usman
5. Derrick Lewis
6. Colby Covington
7. Ronda Rousey
8. Tony Ferguson
10. Marlon Moraes
11. Jorge Masvidal
12. Carlos Condit
13. Anderson Silva
14. Yair Rodriguez
15. Rick Glenn
16. Junior Dos Sanots
17. Josh Barnett
18. Damian Grabowski
19. Mark Coleman

For both the red and the blue corner, recommending user1 fighters:

Carlos Condit	3.893975204534959
Anderson Silva	3.847490141942545
Josh Barnett	3.8407720322685917

Table 1. SVD Results for User 1 Recommendations

Results for ALS by Takedown Percentage:

For the Blue Corner and Red Corner Respectively:

Fighter	Score
Chris Gutierrez	0.056119
Marcos Vinicius	0.051951
Vince Morales	0.050364
Song Yangdon	0.050241
Walel Watson	0.041870
Khalid Taha	0.039220
Tim Gorman	0.037402
Ian Loveland	0.036012
Damacio Page	0.034774
Ian Entwistle	0.031591

Table 2. ALS For Blue Corner

Fighter	Score
Alex Caceres	0.230035
Brad Pickett	0.119081
Lucas Martins	0.081333
Edwin Figueroa	0.078986
Benito Lopez	0.068402
Johnny Eduardo	0.060121
Dustin Pague	0.056784
Chris Beal	0.055494
Nathaniel Wood	0.054822
Wanderlei Silva	0.051620

Table 3. ALS for Red Corner

5. Discussion

Firstly, the project suffers from the commonly known as a "cold start" problem, in which case the recommendation system is provided insufficient information about its users or items. The data presented does not allow for a recommendation system to draw such inferences because the data given does not contain about viewers. Specifically, the engine suffers from a case known as "Systematically Bootstrapping", commonly referred to as "The New Community Case", in which the data has virtually no information the engine can rely upon is present. Often this refers to having completely new users and completely new items. Our data, however, does not have columns for users or ratings. Therefore, commonly known strategies to implement implicit or explicit feedback systems have to be modified to fit our current data have been proven difficult. Gathering a clear understanding the data allowed us to circumvent some of these difficulties and allowed us to implement three different commonly used algorithms for recommendation systems as mentioned earlier in the methods portion. As mentioned, the algorithms have been analyzed by separating out the blue and red corners. The follow provides discussion based on our results after filtering both the red and blue corner.

PCA Analysis

Looking into the PCA implementation, the elbow point from the scree plot suggests to use 2 for the number of components to help visualize the data. The principle comments scatterplot selects the winner, also allowing for a draw, and plots. In this dataset, all of the clusters overlap and do not form distinct groups. This suggests that there is no distinction between the three groups. The source of variation is similar between the three, application of a heat map included within the appendix, that some pair-wise features have scaling significant relevance.

k-Means Analysis This implementation further illustrates the results previous found in PCA but shows that clustering in a much more distinct manner. It verifies the results from PCA, shows a non-linear relationship between the features and the scattered nature of the dataset.

Linear Regression Analysis

As proposed by the PCA reduction, there may be little distinction amongst the features, but in order to further explore this and verify, we implemented linear regression amongst the different pair-wise features. As presented by the methods section, two fairly good fits pertain to the total number of rounds a fighter per corner is able to endure. These instances produce decently well fitted lines, with mean squared error, while not close to 0, much smaller than any number larger than 2 with low variance. However, this is not indicative of the data, as suggested by the PCA reduction.

On average, 86 percent of data points fall on the regression line according to the average coefficient of determination or its R squared value. However, these plots have a high mean squared average. This indicates a high variance and high bias estimate. However, much of the data overall is scattered wildly around the regression line or vaguely clustered towards on section of the graph. This suggests that are no specific pair wise feature to center around a recommendation system, given that no feature in particular best ensures a fighter to win. This, however, does prevent making recommendations with just one feature. This suggests that this is no optimal feature to center the engine around, but it is possible to produce similar fighters based on one feature. For example, it is still noteworthy to recommend other grapplers based on their takedown or submission percentages.

Furthermore, with working with a cold start problem, it is ineffective to simply allow users to initially see fighters that have a successfully executed a particular feature– since no distinct feature contributes heavily to a win. This resonates with the concept of implementing many different striking and grappling styles to win.

k-Nearest Neighbors Analysis

The implementation of k-Nearest Neighbors (kNN) provides an outline of simply recommending fighters that are numerically similar to each other. No feature is weighed more than other, so implementing a kNN on each fighters attributes will find fighters most similar to each other, which is the expected output. This requires the user to know at least one fighter and it is not prompted a selection amongst fighters, since the function can be used multiple times on individuals fighter. A fighter's stance is ignored in this implementation since fighters are capable of switching stances throughout the match. Note, because of how the data is has originally been structured, the results produce a pair of fighters that fought similarly to the recommended person in a specific match.

In the blue corner, for Valentina Shevchenko, the current female's flyweight division champion, only two recommendation based on one bought: Cortney Casey and Cristina Stanciu. However, for recommendations that are male, there are a total of six distinct male fighters that fight similarly to Schevchenko on a specific bout – four of those male fighters fought in a the bantam weight class which is one weight class above flyweight.

In the red corner, for Schevchenko, only two fighters, Amanda Nunes and Ronda Rousey in UFC 207, both fighters being the current and previous bantam weight champion.

Since kNN classifies objects based on their closest training examples and is a variation of instanced based learning, the computation is delayed until the actual moment for classification. For the red corner, for example, using kNN recommends other female fighters who also hold the championship belt. It also suggests that the red corner may have more champions, even though by definition the corners only colloquially indicate which side a fighter will fight on. In higher dimensions, however, this method would prove ineffective since determining the value of k will become more increasingly difficult. Therefore, simply recommending fighters based on other fighter similarity during one particular match may not be the most efficient method to recommend fighters. The accuracy of this implementation is relatively high, and therefore can be assumed, even though simple, effective with lower dimensions.

Furthermore, another drawback of this implementation is the structure of the dataset. It does not provide how a fighter fights on average but rather on one specific match. The results of kNN do not fully accurately determine the best fighter that could be recommended because it's based on one instance.

Singular Value Decomposition

The singular value decomposition (SVD) method implemented is an extension of the PCA method but begins to address the cold start problem originally proposed. As opposed to modifying the original dataset with more columns filled with random ratings, and rates from 1-5 as a standard scale. The SVD runs appropriately by R being the matrix with rows being users and columns being ratings for a fighter.

Creating this R matrix introduces the possibility of collaborative filtering on a dataset that does not originally have this explicit input. For a user with very little knowledge on any particular fighter, they can refer to other people's interests to make a decision. Now, the problem only relies on at least one user having a partially full matrix to share their interests with others. R may be sparse, but it begins to start a hybrid recommendation system. Reducing the cold start problem may just force one user to know many different fighters, but it is still improvement over having no data at all to operate on.

Running SVD on both the red and the blue corner produced the same recommended fighters with the same ratings – indicating that the recommendations are based only off a user's preference for a fighter and disregards any of that fighter's attributes. This drastically differs from kNN which is entirely dependent on a fighter's performance in the cage. This is the clear distinction between a content based recommendation system and a collaborative based recommendation system.

Alternating Least Squares

The implementation for Alternating Least Squares (ALS) not only recommends fighters based on matrix factorization by also gives users the ability to choose a particular feature they may be interested in. The differs from the SVD implementation which is dependent on user input. AlS minimizes the Square error over all the observed entries, in this case each fighter's features, while keeping other factors fixed by optimizing over U and V matrices. This requires a training method, and it more reliable and efficient in larger dimensions than kNN is because of the optimization and minimization properties. It, however, does not address the cold start problem, but it offers the users another route to consider. Instead of requiring the user to have prior knowledge of the fighters, they would have to have prior knowledge to at least some aspect of MMA or fighting.

For our data, we implemented ALS on a single feature, takedown percentage, as example. While the intention of the recommendation engine is not to filter by gender specifically, this highlights a large drawback of using only one feature to recommend. Firstly, the female's data is sparse and thus would not filter as dominantly as the men's would. Secondly, takedown percentage takes into account the total amount of attempted takedown and the amount of successfully landed takedowns. In order to look at takedowns, as a whole, the engine must take into consideration overall volume. In a large dataset, only recommending based on one feature cannot successfully encapsulate the fighter's similarity to another.

However, this implementation can be seen as an extension of kNN, in which instead of finding similar fighters within some distance, it finds similar fighters based on the takedown percentage. With further exploration, reduction of the set into grapplers and strikers, this could be used more effectively to determine a similar fighter.

6. Conclusion

Three different types of recommendation engines were produced for this project: content-based filtering, collaborative filtering using explicit feedback, and collaborative filtering using implicit feedback. Respectively, those reference k-Nearest Neighbors, Single Variable Decomposition, and Alternating Least Squares. No algorithm was completely successful at addressing the cold start problem, but, if the dataset has no explicit or implicit feedback, then starting a recommendation system may be difficult.

However, each implementation highlighted key aspects about creating a recommendation system for a dataset without feedback. kNN can accurately recommend fighters soley based on features but does not size well with higher dimensions. SVD requires user input but it accurately filters recommendations based on other people's input. ALS can filter recommendations by a specific features.

7. Future Work

For further work, for SVD, it may be useful to implement SVD by scraping user ratings from various websites into a dataset and directly integrating this into the working dataset. For kNN, it would be useful to implement different methods variations and test different values for k. For ALS, it would be important to evaluate performance of ALS using Mean Precision Analysis, Serendipity and Novelty Calculations. With further exploration of these topics, building a hybrid recommendation engine combining these concepts could potentially allow other implementations on datasets without explicit or implicit feedback.

8. References

Aggarwal, Karmanya. "UFC Fight Data". *UFC Fight Data*. Kaggle, 11 Dec 2018, https://www.kaggle.com/calmdownkarm/ufcdataset.

B, A. (2016, June 13). Recommender Systems - It's Not All About the Accuracy. Retrieved April 12, 2020, from https://gab41.lab41.org/recommender-systems-its-not-all-about-the-accuracy-562c7dceeaff

Victor. (2018, July 10). ALS Implicit Collaborative Filtering. Retrieved April 9, 2020, from https://medium.com/radon-dev/als-implicit-collaborative-filtering-5ed653ba39fe

Li, S. (2019, April 22). Building a Collaborative Filtering Recommender System with ClickStream Data. Retrieved April 11, 2020, from https://towardsdatascience.com/building-a-collaborative-filtering-recommender-system-with-clickstream-data-dffc86c8c65