A Comparison Between Collaborative, Content-Based and Hybrid Recommender Systems

BASEL ELZATAHRY, Davidson College, USA
NATNAEL MULAT, Davidson College, USA
PAUL CHOI, Davidson College, USA
RIDA SHAHID, Davidson College, USA

Recommender systems play vital roles in many online platforms to ensure the best experiences for users by personalizing their preferences. Since there are multiple constraints in real life application, there are trade offs made between accuracy and coverage to produce the optimal result for recommendations. In this research paper, we have considered multiple factors–accuracy, coverage, p-values, and t-test–to determine the best recommender system algorithm by conducting different trials with varying parameters.

Additional Key Words and Phrases: Recommender Systems, Content-based Filtering, Hybrid Recommenders, Similarity and Prediction techniques

ACM Reference Format:
Basel Elzatahry, Natnael Mulat, Paul Choi, and Rida Shahid. 2021. A Comparison Between Collaborative, Content-Based and Hybrid Recommender Systems. 1, 1 (May 2021), 13 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

There are different types of content-based algorithms for recommendation systems, such as Feature Encoding recommender algorithms which acquire item features for given items with content similar to the user's preference, and Term Frequency–Inverse Document Frequency (referred to as TF-IDF in this paper) recommender algorithms which encodes words (terms) in the Vector Space Model in order to summarize the content of a document. In our prior analysis of recommender systems, we compared different collaborative filtering algorithms, User-based and Item-based, and also different similarity methods, Euclidean Distance and Pearson Correlation [1],[2]. We also utilized Matrix Factorization methods such as Stochastic Gradient Descent (SGD) and Alternating Least Squares (ALS) to predict ratings [3].

In order to expand upon the algorithms we have previously developed, we consider Content-based and Hybrid recommenders with different variations and run the program on the data set "ML-100K" [4]. We designed two content-based filtering algorithms, Feature Encoding and TF-IDF, and hybrid recommenders which combined collaborative and content-based methods. The two important variations we added, in addition to the algorithmic approach, are hybrid weighting for hybrid systems and similarity threshold for TF-IDF. We decided to add these variations aiming to see if

Authors' addresses: Basel Elzatahry, Davidson College, 102 N Main Street, Davidson, North Carolina, 28036, USA, baelzatahry@davidson.edu; Natnael Mulat, Davidson College, 102 N Main Street, Davidson, North Carolina, 28036, USA, namulat@davidson.edu; Paul Choi, Davidson College, 102 N Main Street, Davidson, North Carolina, 28036, USA, pachoi@davidson.edu; Rida Shahid, Davidson College, 102 N Main Street, Davidson, North Carolina, 28036, USA, rishahid@davidson.edu.

there is a variation in the content-based or hybrid recommender system that yields a more accurate model compared to collaborative filtering systems.

The primary goal of this research paper is to determine the best set of variations among eleven possible combinations of hyperparameters for content-based and hybrid recommender system algorithms. Next, we compare the best variations from each algorithm (TF-IDF, Feature Encoding or Hybrid) with the best variations from Item-based, User-based, SGD and ALS models. We believe that our findings will serve as the basis for future research and study of recommender systems. Our findings can be applied to other data sets of a similar size and data structure to ML-100K [4]. This will allow future researchers to efficiently generate accurate recommendations with high coverage and determine which recommender system to use among collaborative, content-based and hybrid recommenders.

2 RELATED WORK

Our method of matrix factorization to approximate the original user-item matrix is derived from the works of Koren et al [3]. Specifically, the approach is to minimize the regularized squared error on the set of known ratings by using stochastic gradient descent and alternating least square methods [4]. The works of Pazzani et al. have inspired us to explore the TF-IDF algorithm and cosine similarity of vectors to generate a content-based recommender system [5]. As an alternative approach, we would like to compare the results of TF-IDF with feature encoding. Our implementation of the hybrid recommender is based on the work of Adomavicius et al and R. Burke [6],[7]. The purpose of this implementation is to minimize the limitations of collaborative and content-based filtering.

In order to determine the best result, along with accuracy, we have considered the coverage of items for each set of variations. Our definition of coverage comes from Herlocker et al., who define coverage to be the "number of items" for which evaluations can be made out of the total number of items recommendations were requested for [2]. In order to combine both accuracy and coverage, we referred to the work of Seminario and Wilson, who use the "AC measure" as a single metric for determining the best results [8].

3 THESIS

3.1 Purpose

The purpose of this research paper is to determine which recommender system and its hyperparameters yield the best results. As we have tested different collaborative filtering recommendation algorithms in our previous research, the primary focus of this research is to test different sets of hyperparameters for content based and hybrid recommender systems and make an overall comparison. We define the best recommender system based on accuracy and coverage and determine whether to accept or reject our hypothesis based on the p-values and t-tests [9].

3.2 Motivation

The main motivation of this research lies in our desire to determine the best recommender system by conducting our own experiment because it is more helpful to comprehend the mechanism and performance of the recommender systems if we design the experiments and control the hyperparameters. This research would give better insight of the recommender systems and allow us to analyze the data more effectively before reaching the conclusion. Moreover, since we have determined the best set of hyperparameters with collaborative filtering in our previous research, we aim to further analyze our results by comparing the best results using various algorithms. In summary, this work seeks

to determine the best algorithm and its corresponding set of parameters for the ML-100K dataset with an intent of applying the same parameters to other similar datasets in future for efficiently generating recommendations.

3.3 Hypotheses

Based on previous research, we hypothesize that the hybrid recommender system will perform the best [7]. It is known to minimize the limitations of content-based and collaborative systems and thus, generate more accurate recommendations with a larger coverage [7].

In terms of the similarity metric, our previous work showed that more accurate ratings were predicted using Pearson Correlation as compared to Euclidean Similarity Distance for RS in collaborative filtering. This was shown by the error and AC Measure which were lower with Pearson Correlation than with Euclidean Distance. Therefore, we hypothesize that Pearson Correlation will prove to be more accurate than Euclidean Distance for the hybrid system as well.

Moreover, we hypothesize that the best results will be with a hybrid weight of 1 as we predict that it is nevertheless better to take into full account of the similarity generated from a recommendation rather than arbitrarily defining the weight by ourselves.

Thus, we hypothesize that our best performing model will be a hybrid recommender system using Pearson Correlation with a hybrid weight of 1.

3.4 Approach

Our approach is to implement feature encoding by obtaining item features for each item; i.e. genres for movies. Then, we use the feature profile of the items and user's preference matrix to calculate the weighted preference vector. The user's preference matrix is based on the movies the user has already seen and rated. Finally, we generate recommendations for unrated items by taking the sum of the product of each weighted normalized preference vector and respective average user score for the feature.

We implement TF-IDF to evaluate the features for all the items in the U-I matrix [9]. This is done by calculating the frequency of keyword i in document j divided by the number of terms in document j (TF) [6]. However, we need to consider that some words may appear more than other words simply because it is a commonly used term. Hence, we have to factor how common the words i is by taking the the inverse document frequency (IDF):

$$IDF(i) = \log \frac{N}{n(i)} \tag{1}$$

where, N is the total documents and $n(i)$ is the number of documents from N in which keyword i appears [6]. Finally, we multiply TF with IDF to get weighted word value [6]. We add each weighted word value into a matrix to eventually render a word vector for every item [6]. To generate recommendations, we calculate the similarity of word vectors in Vector Space Model (VSM) using Cosine Similarity [6]. This is calculated based on the angle between vectors as:

$$\textbf{cosine sim}(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| * |\vec{b}|} \tag{2}$$

The similarity threshold for TF-IDF is the lower limit of the similarity values to consider.

Thirdly, we develop our hybrid recommender system by combining our best collaborative filtering model, based on previous research, with TF-IDF. For the collaborative filtering, we choose Item-based filtering with 100 neighbors and a significance weight of $\frac{n}{50}$ using both, Pearson Correlation and Euclidean Distance for RS [2]. Just for comparison, we also implement some variations with a significance weight of $\frac{n}{1}$. For TF-IDF, we choose a similarity threshold of > 0.

When the TF-IDF similarity is zero, we replace the similarity with the item-item similarity matrix value for that specific item-item combination multiplied by the hybrid weight.

Next, for each of our models, we predict ratings using Leave-One-Out Cross-Validation (LOOCV) and calculate the errors, as described in our previous work. Similarly, we measure the accuracy of our models based on the mean squared error, root mean squared error and mean absolute error. A lower error represents a higher accuracy. We also calculate the coverage and AC Measure of each variation and define our best model to be the one with the lowest AC Measure [8]. In AC Measure, we use both accuracy and coverage by dividing the first by the second, meaning that the lower the AC Measure the better, since we need to maximize the coverage and minimize the error [8]. For further details regarding the performance metrics, please refer to our previous work.

Finally, we perform a test of hypothesis on our accuracy by calculating p-values and performing the t-test [9]. This is done in order to ensure that there is a statistically significant difference between different variations and the results are not due to luck [9]. The t-test is performed as follows:

$$t = \frac{\overline{x_1} - \overline{x_2}}{\sqrt{(s^2(\frac{1}{n_1} + \frac{1}{n_2}))}} \tag{3}$$

where, t is the t-value, x_1 and x_2 are the means of the two variations being compared, s^2 is the pooled standard error of the two variations, and n_1 and n_2 are the number of ratings for each variation. A p-value greater than the threshold and absolute value of t-test less than the critical t-value indicate that we fail to reject the null hypothesis and the difference between the means of the two variations is due to chance [9].

4 EXPERIMENTAL DESIGN

4.1 Datasets and Algorithms

The data used in this study was the MovieLens 100K Dataset downloaded from GroupLens Research [4]. MovieLens is a web-based research recommender system run by GroupLens Research at the University of Minnesota. The ML-100K data set consists of 100,000 ratings for 1,682 movies and 943 users with each user having rated at least 20 movies. Throughout this paper, we refer to this data set as ML-100K. Ratings provided in this data set consist of integer values between 1 (did not like) to 5 (liked very much).

For TF-IDF, we considered similarities greater than 0, the 25^{th}, 50^{th} and 75^{th} percentiles by varying the threshold between >0, 0.25, 0.50 and 0.75. These values were chosen based on the distribution of cosine similarities as seen in Fig. 1, their mean (0.5750) and standard deviation (0.2696). We also implemented Feature Encoding as described previously. Lastly, we implemented a hybrid algorithm by varying the hybrid weight between 0.5 and 1 in order to observe the effect of using the actual item-item similarity versus half the similarity. We predicted that a larger weight would not have a significant enough difference and a smaller weight would diminish the similarity more than needed. We also vary the similarity method for hybrid systems between Euclidean Distance for RS and Pearson Correlation. All the predictions were generated using Leave-One-Out Cross-Validation (LOOCV).

4.1.1 Test cases:

In order to test the overall hypothesis, the following test cases were developed and executed using the ML-100K dataset:

1. TF-IDF, Similarity Threshold > 0
2. TF-IDF, Similarity Threshold > 0.25
3. TF-IDF, Similarity Threshold > 0.50

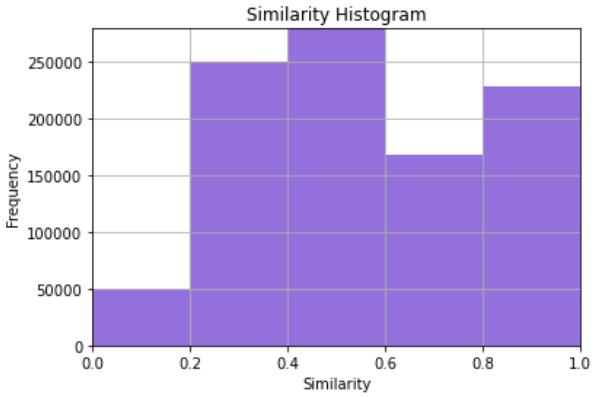


Fig. 1. Distribution of Cosine Similarities.

4. TF-IDF, Similarity Threshold > 0.75
5. Feature Encoding
6. Hybrid, Pearson Correlation, Hybrid Weight = 1, Significance Weight= $\frac{n}{50}$
7. Hybrid, Euclidean Distance, Hybrid Weight = 1, Significance Weight= $\frac{n}{50}$
8. Hybrid, Pearson Correlation, Hybrid Weight = 0.5, Significance Weight= $\frac{n}{50}$
9. Hybrid, Euclidean Distance, Hybrid Weight = 0.5, Significance Weight= $\frac{n}{50}$
10. Hybrid, Pearson Correlation, Hybrid Weight = 1, Significance Weight= $\frac{n}{1}$
11. Hybrid, Euclidean Distance, Hybrid Weight = 1, Significance Weight= $\frac{n}{1}$

4.1.2 Accuracy and Coverage Metrics:

We used MSE, MAE and RMSE to measure the accuracy of the rating predictions. For measuring the coverage, we calculated the percentage of the dataset for which the recommender system was able to provide predictions (using LOOCV). We then calculated AC Measures for all test cases using each accuracy metric.

4.1.3 Test of Hypothesis:

In order to determine the significance of the difference between our best variations, we performed the t-test using the p-values and t-statistics values. The significance values we compared our results with are 0.1, 0.05, and 0.01, for the t-statistics we tested our hypothesis on 90%, 95%, and 99% confidence interval using the t-statistics tables to find the respective critical values for a two-tailed hypothesis test. Please note that the hypothesis mentioned in this paper test:

- H_0: $\mu_1 = \mu_2$
- H_1: $\mu_1 \neq \mu_2$

Where μ is the expected value for an independent samples of scores In addition, it should also be noted that the test assumes that the populations have identical variances by default.

4.1.4 Variations Comparison:

We compared the best Feature Encoding, TF-IDF and Hybrid recommender variations with our best variations for collaborative filtering using user-based methods, item-based methods, matrix factorization using stochastic gradient descent, and matrix factorization using alternating least squares. For item-based, we chose the variation with Pearson

Correlation as the similarity method, significance weight of $\frac{n}{50}$ and a similarity threshold of 0 [10]. For user-based, we chose the variation calculated using Pearson Correlation with a significance weight of $\frac{n}{25}$ and a similarity threshold of 0. For matrix factorization with ALS, we used 2 features, a regularization constant of 0.01 and 20 iterations. Lastly, for matrix factorization with SGD, we used 200 latent features, a learning rate of 0.02, and a regularization constant of 0.002.

5 RESULTS

Before evaluating our results, we find it important to understand the data that we are working with so the results can be more digestible and easy to interpret.

5.1 Descriptive Analytics for ML-100K:

```
Number of users: 943
Number of items: 1664
Number of ratings: 99693
Overall average rating: 3.53 out of 5, and std dev of 1.13
Average item rating: 3.08 out of 5, and std dev of 0.78
Average user rating: 3.59 out of 5, and std dev of 0.44
User-Item Matrix Sparsity: 93.65%
```

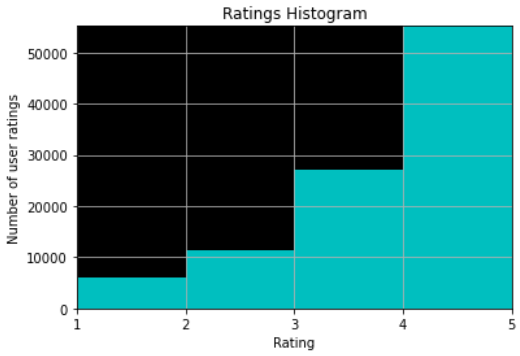


Fig. 2. Histogram of the Ratings in ML-100K.

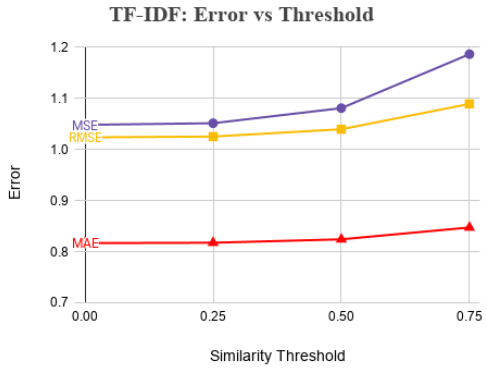
For further details regarding the data, refer to our previous work.

5.2 MSE, MAE, RMSE Results:

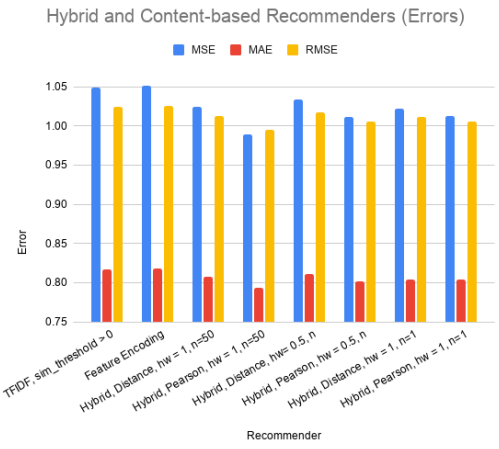
As mentioned above, we use different algorithms and parameters resulting in 11 total variations. The results for the accuracy measures we have used, in order to determine the best variation, are shown in Fig. 3. We also find that the difference between feature encoding, and TF-IDF with similarity thresholds > 0 and > 0.25 is due to luck and we fail to reject the null hypothesis for them. The difference between the remaining variations is signnificant.

The overall trends in accuracy are:

1. For TF-IDF, as we increase the similarity threshold, the error increases which means the accuracy decreases.



(a) TF-IDF Results with Different Thresholds.



(b) Errors for Hybrid Recommenders compared to FE and best TF-IDF variation. n is the signficance weight for item-item filtering and hw is the hybrid weight.

Fig. 3. Accuracy results.

2. Feature encoding is more accurate than TF-IDF with similarity thresholds > 0.5 and > 0.75 and less accurate than TF-IDF with similarity thresholds > 0 and > 0.25. However, based on the tests of hypothesis, feature encoding, and TF-IDF with similarity thresholds > 0 and > 0.25 are equal.
3. Hybrid recommender with Pearson similarity, a significance weight of $\frac{n}{5}$ and a hybrid weight of 1 has the lowest errors among all content-based and hybrid recommenders, confirmed using the p-values and t-test.

5.3 Coverage and AC Measure Results:

The effect of varying the similarity threshold for TF-IDF on coverage and AC Measure can be seen in Fig. 4. The overall coverage and AC Measure comparison can be seen in Fig. 5.

The observed trends regarding TF-IDF are:

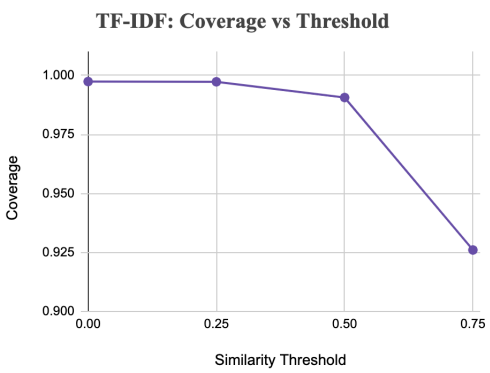
1. Increasing the threshold decreases the coverage and increases the error.
2. The lower the threshold, the lower the AC Measure and vice versa.

The general trends for the hybrid systems are:

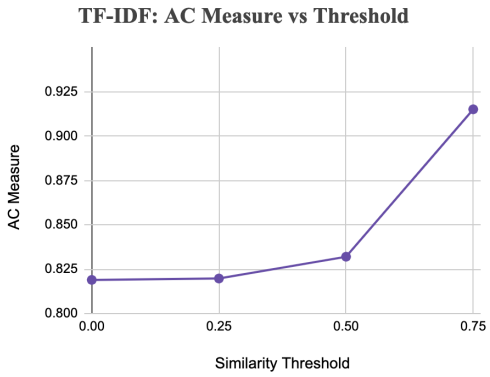
1. Pearson correlation is more accurate and has more coverage than Euclidean distance.
2. Hybrid weight of 1 yields a similar coverage to a hybrid weight of 0.5. However, hybrid weight of 1 has a better accuracy which yields a better AC Measure.
3. n = 50 has about the same coverage as n = 1, but it has a better accuracy and a better AC, accordingly.

The key findings of these results are:

1. TF-IDF with a similarity threshold > 0.5 has the highest error, lowest coverage and highest AC Measure and thus, the worst overall performance.

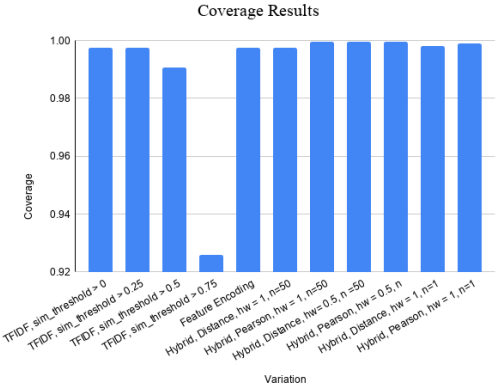


(a) Coverage trends with varying similarity thresholds .

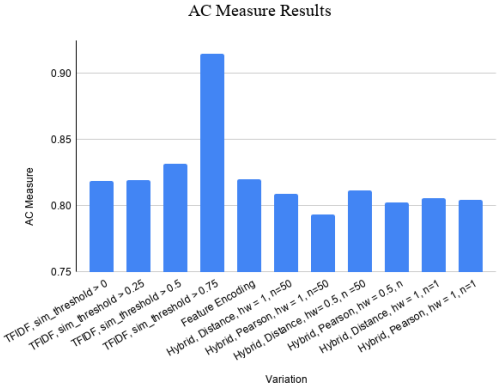


(b) AC Measure trends with varying similarity thresholds.

Fig. 4. TF-IDF results.



(a) Coverage comparison between all variations.



(b) AC Measure comparison between all variations.

Fig. 5. Comparison between all tested variations. “hw" represents hybrid weight.

2. Hybrid with Pearson Correlation, significance weight of $\frac{n}{50}$ and hybrid weight of 1 has the lowest error, highest coverage and lowest AC Measure and thus, the best overall performance.
3. Hybrid models perform better than TF-IDF and Feature Encoding as they generally have lower errors, higher coverages and lower AC Measures.

5.4 Best Results Comparison:

Finally, we compare the best variations from all the implemented algorithms: Matrix Factorization with Stochastic Gradient Descent (SGD) and Alternating Least Squares (ALS), User-User (UU), Item-Item (II), TF-IDF, Feature Encoding (FE) and Hybrid.

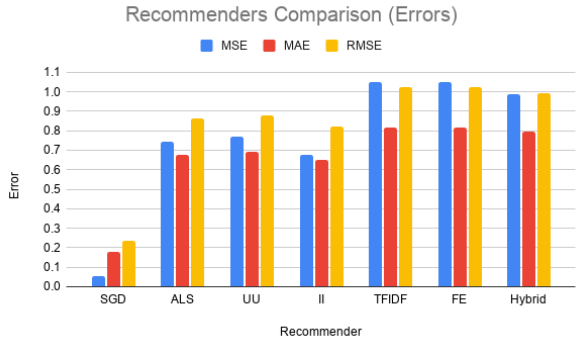
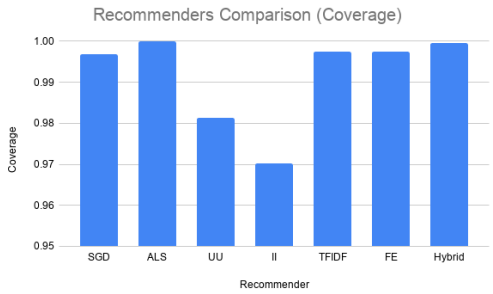
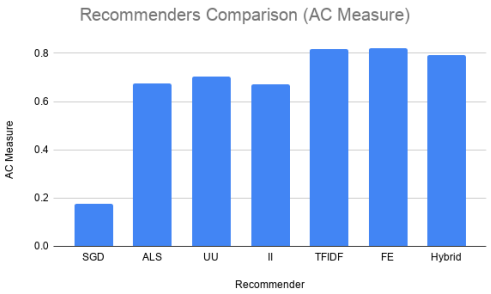


Fig. 6. Overall accuracy comparison between all best models.



(a) Coverage comparison between best variations from all methods.



(b) AC Measure comparison between best variations from all methods.

Fig. 7. Overall coverage and AC comparison between all best models.

The key findings of our overall research are:

1. SGD results in a significantly lower error than all other methods but this model is most likely overfitting.
2. ALS results in the highest coverage.
3. Collaborative models have higher accuracies than content-based and hybrid models.
4. User-based and item-based methods have the lowest coverages.
5. Collaborative filtering performs better than content-based and hybrid methods as collaborative models have lower AC Measures.

5.5 Test of Hypothesis:

We have done test of hypothesis to compare similar results and will expand more on its usage in the discussion section. We compared our TF-IDF, FE, and Hybrid; we also compared our top 4 results from all the experiments we have tried this semester including MF-ALS, User-User and Item-Item.
The key findings of our test of hypothesis are:

P-VALUES

	TFIDF, > 0	TFIDF, > 0.25	TFIDF, > 0.5	TFIDF, > 0.75	FE
TFIDF, > 0		0.6419706946	2.32E-06	4.81E-74	0.622548806
TFIDF, > 0.25	0.6419706946		2.06E-05	1.79E-70	0.9780800528
TFIDF, > 0.5	2.32E-06	2.06E-05		7.77E-42	2.34E-05
TFIDF, > 0.75	4.81E-74	1.79E-70	7.77E-42		3.20E-70
FE	0.622548806	0.9780800528	2.34E-05	3.20E-70	
Hybrid, distance, hybrid_weight = 1, n=50	0.000277021830	4.22E-05	1.06E-16	2.98E-103	3.78E-05
Hybrid, Pearson, hybrid_weight = 1, n=50	1.09E-19	1.59E-21	2.21E-42	5.98E-156	1.28E-21
Hybrid, distance, hybrid_weight = 0.5, n =50	0.0282639379	0.007926168602	5.90E-12	3.73E-91	0.00733068696
Hybrid, Pearson, hybrid_weight = 0.5, n =50	5.697279648	7.61E-10	5.31E-25	4.26E-122	6.52E-10
Hybrid, distance, hybrid_weight = 1, n=1	0.000277021830	4.22E-05	1.06E-16	2.98E-103	3.78E-05
Hybrid, pearson, hybrid_weight = 1, n=1	1.09E-19	1.59E-21	2.21E-42	5.98E-156	1.28E-21

(a) p-values

T-Test

	TFIDF, 0	TFIDF, 0.25	TFIDF, 0.5	TFIDF, 0.75	FE
TFIDF, sim_threshold > 0		-0.4649459171	-4.723058572	-18.21170651	-0.4922420
TFIDF, sim_threshold > 25th percen	-0.46494591		-4.258549322	-17.75515226	-0.0274760
TFIDF, sim_threshold > 50th percen	-4.72305857	-4.258549322		-13.5546795	4.2295490
TFIDF, sim_threshold > 75th percen	-18.2117065	-17.75515226	-13.5546795		17.722553
FE	-0.49224201	-0.02747607082	4.229549084	17.72255349	
Hybrid, distance, hybrid_weight = 1,	3.63595538	4.094966905	8.298520844	21.58961526	4.1204732
Hybrid, Pearson, hybrid_weight = 1,	9.080402774	9.529899585	13.64675254	26.62928245	9.5525471
Hybrid, distance, hybrid_weight = 0.	2.193619303	2.655225543	6.882558805	20.25839613	2.6814637
Hybrid, Pearson, hybrid_weight = 0.	5.697279648	6.153237521	10.32863823	23.51504466	6.1777083
Hybrid, distance, hybrid_weight = 1,	3.63595538	4.094966905	8.298520844	21.58961526	4.1204732
Hybrid, pearson, hybrid_weight = 1,	9.080402774	9.529899585	13.64675254	26.62928245	9.5525471

(b) t-test

Fig. 8. p-values and t-Test for TFIDF, FE, and Hybrid comparison. The light purple and cyan values show the statistically insignificant results.

1. TF-IDF of threshold 0 and threshold 0.25 are similar since we were unable to reject the null hypothesis implying that they have the same results.
2. The best TF-IDF model and FE have a p-value and t-tests of 0.62 meaning that we fail to reject the null hypothesis that the variations have the same result of means.
3. When we compared our top 4 models, their p-values and t-tests rejected the null hypothesis and showed that they are different meaning that it is easy to differentiate between them using our error and coverage metrics.

P-VALUES

	Item-Item	User-User	MF-ALS	Hybrid, pearson, 1 ,50
Item-Item		7.79E-79	3.38E-45	0.00E+00
User-User	7.79E-79		1.49E-06	5.45E-298
MF-ALS	3.38E-45	1.49E-06		0.00E+00
Hybrid, pearson, 1 ,50	0.00E+00	5.45E-298	0.00E+00	

(a) p-values

T-TEST

	Item-Item	User-User	MF-ALS	Hybrid, pearson, 1 ,50
Item-Item		-18.806856	-1.41E+01	-5.64E+01
User-User	-18.8068563		4.81E+00	-3.70E+01
MF-ALS	-1.41E+01	4.81E+00		-4.22E+01
Hybrid, pearson, 1 ,50	-5.64E+01	-3.70E+01	-4.22E+01	

(b) t-test

Fig. 9. p-values and t-test for top 4 variations.

6 DISCUSSION

6.1 Feature Encoding vs. TF-IDF

As shown in the results section, TF-IDF with a similarity threshold of greater than 0 and greater than 0.25 has a slightly higher accuracy (less error) than FE. Due to the close values of the errors, we refer to the p-values and t-tests between these variations which and find the p-values to be higher than the significance value of 0.1, 0.05, and 0.01, similarly, the t- value of those variation is lower than the critical values of 0.1, 0.05, and 0.01. Thus, we fail to reject the null hypothesis that the variations have the same result of means. This indicates that the difference between the errors is not statistically significant, which is not helpful in determining which model is better. Moreover, considering these models have the same coverage, their AC measure is really close as well which shows that these models have the same performance.

In summary, our results for whether FE or TF-IDF is better were inconclusive based on the test of hypothesis. However, we know that TF-IDF with thresholds greater than 0 and greater than 0.25 performed slightly better than FE,

whereas FE performed better than TF-IDF when TF-IDF had higher thresholds of 0.5 and 0.75. Although we varied the similarity threshold under the premise that higher threshold would give more accurate result, our results indicated that is actually better to take into account of the lower similarity for recommendation rather than dismissing the lower similarity values altogether. Excluding these ratings negatively impacted the accuracy of our TF-IDF models.

6.2 Hybrid vs. Content-Based Systems

As we can see from the result section, the hybrid recommenders proved to be superior to the content-based recommenders. The best hybrid model was one with Pearson Correlation, significance weight of $\frac{n}{50}$, and a hybrid weight of 1. This model had the lowest MSE (0.989) and the highest coverage (99.96%). Subsequently, the AC measure was 0.79, which was the lowest AC measure in our current experiments. However, when we compared it to best TF-IDF and FE models, we used the p-values and found them to be less than 0.05, meaning that it rejected the null hypothesis that hybrid methods were better than content-based methods. Moreover, the best performance with a hybrid weight of 1 showed that including the actual similarity proved to be the most accurate and adding weight misrepresented the similarity value.

6.3 Hybrid and Content-Based vs. Collaborative Systems

Overall, we observed that collaborative filtering generated more accurate recommendations than content-based and hybrid filtering. On the other hand, content-based and hybrid recommenders had higher coverage than item-based and user-based models. The matrix factorization models had some of the lowest errors and highest coverage, but because these models may be overfitting the data, we were reluctant to state that the it did indeed give the best results. Moreover, we observed that hybrid models improved the performance of content-based models, but still, collaborative models outperformed the hybrid models. We believe that this was because movies were based on nuances that content based methods failed to consider. Our hybrid system mostly adapted TF-IDF model with only the missing similarities filled by similarities from the collaborative methods. This demonstrated that the contribution of collaborative methods was not as significant in our hybrid models as we anticipated. In conclusion, collaborative models have the lowest AC Measures and thus, perform better than content-based and hybrid models.

7 CONCLUSION

Although we have found that item-item collaborative filtering system performed the best overall, in this particular research where we conducted experiments on content-based and hybrid models, we concluded that hybrid models were better than TF-IDF and FE models. Following the lead were TF-IDF models of > 0 and > 0.25, and FE model. This is the ranking that summarizes our analysis:

1. Hybrid Models ((Hybrid Weight = 1, Significance Weight = $\frac{n}{50}$, Pearson) is the best among content-based and hybrid models)
2. FE = TF-IDF > 0 = TF-IDF > 0.25
3. TF-IDF > 0.5

However, in terms of performance, FE was notably faster than the TF-IDF in predicting results when we ran the datasets. There are always trade-offs in these recommender systems, but for people who prefer performance over accuracy, we believe FE would be a better choice.

We hypothesized that hybrid method would perform better than both content based and collaborative methods. Although hybrid method was better than content-based method, we observed that collaborative method performed better on its own. Thus, collaborative filtering works best for ML-100K as it predicts the most accurate ratings with the lowest AC Measures.

In future research, we would like to implement other hybrid algorithms such as weighted, cascade and meta-level hybrid recommenders to discern whether those models might improve the accuracy of the hybrid model to the extent of which they are better than collaborative recommender systems[9]. Moreover, we should like to implement the recommender models on a larger data set such as ML-100M to test if our current observations still hold [4]. Lastly, we would analyze our matrix factorization methods to test overfitting and ensure the validity of the results so that we could make appropriate comparison using the same metrics.

8 ACKNOWLEDGMENTS

We would like to give special thanks to Dr. Seminario for providing resources including but not limited to research papers and important algorithms and suggestions to implement the recommender systems.

9 REFERENCES

[1] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In Proceedings of the ACM SIGIR Conference, 1999.

[2] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. Riedl. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems, 22(1):553, 2004.

[3] Y. Koren, R. Bell and C. Volinsky. Matrix Factorization Techniques for Recommender Systems. In Computer, 42, 8, 30-37 (2009). doi: 10.1109/MC.2009.263., editors, Recommender Systems Handbook. Springer, 2011.

[4] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. GroupLens: an open architecture for collaborative filtering of netnews. In Proceedings of the ACM CSCW Conference, 1994.

[5] Pazzani M.J., Billsus D. (2007) Content-Based Recommendation Systems. In: Brusilovsky P., Kobsa A., Nejdl W. (eds) The Adaptive Web. Lecture Notes in Computer Science, vol 4321. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-72079-9_10

[6] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. In IEEE Transactions on Knowledge and Data Engineering, 17, 6, 734-749 (2005). doi: 10.1109/TKDE.2005.99.

[7] Burke, R. Hybrid Recommender Systems: Survey and Experiments. User Model User-Adap Inter 12, 331–370 (2002). https://doi.org/10.1023/A:1021240730564

[8] C.E. Seminario and D.C. Wilson. Case Study Evaluation of Mahout as a Recommender Platform, July 2012.

[9] G. Shani and A. Gunawardana. Evaluating recommendation systems. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, Recommender Systems Handbook. Springer, 2011.

[10] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In Proceedings of the World Wide Web Conference, 2001.