Capstone Project: Recommender System for Nile-River

Part 1: Plan

1.1 Introduction

The case is that an online retailer (Nile-River) has faced modest increase in sales during back-to-school period. Therefore, Nile-River has to adopt recommender systems to increase the sales to be comparable to other competitors. Our goal is to find 5 products to be displayed as the recommended items. In addition, the recommendation must be based on user profile history. The useful facts to keep in mind are following:

1) Additional sales are divided fairly among three categories of office supplies: school supplies, consumable supplies, and durable office equipment.

2) Most large dollar purchases consist of both small and large purchases.

3) The strength of Nile River is that it has much deeper product catalogs compared to other stores.

1.2 Metrics

- Mean Absolute Error: we need prediction accuracy metric to measure how well this recommender behaves in predicting the ratings

- Discounted Cumulative Gain: since our task is about finding top-5 items, we need measurement in this category to measure how well the algorithm produces top-5 items

- Average Price Standard Deviation: since we want to measure diversity of the recommended items, we need to measure the mix between expensive and cheap items in recommended list in order to recommend small and large items together

- Average Availability: since the strength of Nile River is that it has much deeper product catalog, so we evaluated average availability of top-5 items to do the measurement in this category

1.3 Algorithms

- Content-Based Recommender: Since office supplies products could be bought again by consumers, we would like to offer similar and hopefully better/cheaper to consumers

- User-User Recommender: To add diversity in user's basket, we would like to find what items were bought by similar users in order to recommend these items to that particular users. Items might be in different categories.

- Matrix Factorization: To be able to learn the user behavior and item characteristic using latent variable in MF, this algorithm is the key. Hopefully, we could learn the use behaviors in order to recommend what users may like and purchase.

1.4 Implementation

- Split data into training, validation and testing (60-20-20). Since the data have been trained already, I will only consider the testing part (the last 20%).

- For each of three algorithms,

  a) Compute top-5 list from items that users have rated only to measure mean absolute error and discounted cumulative gain.

  b) Compute top-5 list from any items to compute similarity measure and average availability.

1.5 Hybrid Algorithms

Plan A: Use equal-weighted average of predicted ratings from three algorithms, and pick top-5 items from the list

Plan B: Divide recommendation algorithm, based on the number of rated items. For users with no rated items, use content-based recommendation. For users with some number of rated items, use matrix factorization or user-user recommendation

Plan C: Combine three lists of items such that they minimize average availability

Part 2: Measurement

2.1 Summary Statistics

This table shows the evaluation on each of recommender systems where MAE and DCG are measured by computing top-5 list from items that users have rated only while Average Price Standard Deviation and Average Availability are measured by computing top-5 list from all items. Each metric is evaluated on the last 20 entries of users.

	Content-Based Recommender (CB)	User-User Recommender (UU)	Matrix Factorization Recommender (MF)
1. MAE	0.372	0.333	0.457
2. DCG	16.520	16.865	16.141
3. Average Price Standard Deviation	11.99	19.410	14.977
4. Average Availability	0.575	0.712	0.5196

(Note that if prices are missing in any entries, we compute the standard deviation based on remaining items)

2.2 Evaluation of Algorithms

First, consider prediction accuracy metric, MAE. We needed this measure as a baseline on how each algorithm performs in predicting user ratings. MF results in the lowest accuracy among the three while UU results in the highest accuracy. Second, consider DCG. This measure accounts for how good the top-5 ranking is for each algorithm. MF results in the lowest score while UU results in the highest score (higher means better for DCG). Third, consider average price standard deviation, as a measure of diversity. Since we would like to recommend cheap and expensive items together, this measurement is needed. CB results in the lowest diversity in recommended list while UU results in the highest diversity in recommended list. Finally, consider average availability. Since the strength of Nile-River is deep product catalogs, we would like to offer items which are not easily available in other stores. MF performs best in offering products with low availability while UU performs worst as it offers product with the highest average availability.

To summarize, content-based recommender could not perform well in many categories compared to user-user recommender and matrix factorization. User-user recommender could perform well in many measurements except for utilizing deep product catalogs measured by average availability. Even though matrix factorization did not perform well in some measurements, but it could output product list which is not easily found in other stores.

2.3 Hybrids

Since each measurement has its own strengths and weaknesses, we could combine all three in the hope of achieving better results. Three possible algorithms for hybrid are as follows:

First, from original dataset, we take equal-weighted average of predicted ratings, and pick top-5 items for this list. This is the simplest approach of all three approaches.

Second, we divided three algorithms to perform in area that each could work best. For users with no ratings, we would use content-based recommender as it does not require users' ratings at all. For users with moderate to high number of ratings, we could use either matrix factorization or user-user recommendation. Considering summary statistics on items rated, the mean number of items rated is 14.6 items per user and median is 14 items per user. Hence, we divided consumers into two groups: one group with less than 15 rated items and another group with at least 15 rated items. These two algorithms seems quite opposite to one another: matrix factorization is good at offering products with low availability while not good at offering items with high price variances, while user-user recommender is good at offering items with high price variances, but not good at offering products with low availability. These two goals are essential as recommending low availability item could leverage Nile River strength while recommending products with high price variances could result in large-dollar purchases.

Third, we combined three list of items such that it minimized average availability using greedy approach. This algorithm tries to leverage Nile-River strength in offering low-availability products directly. Then, we could see the impact on price variances of recommended items.

Part 3: Mixing

3.1 Mixing Algorithm

The first option that we could explore is to take equal-weighted average of all three results, and choose top-5 list from that. This is the simplest algorithm, taken as a baseline to be compared with other alternatives.

The second option is segmented recommender. As we have seen in previous parts, user-user recommender tends to be good at prediction accuracy, top-n list accuracy, and providing list with high price variances, but this algorithm is relatively bad at providing list that leverages deep product catalog of Nile-River. However, matrix factorization recommender is not good at prediction accuracy, top-n list accuracy and providing mixes with different prices, but it is good at producing list where average availability of recommended items is low. We explored segmented recommender by dividing users into two groups: people who have rated less than 15 items, and people who have rated at least 15 items. As we could not know what algorithm is most suitable for each group, we decided to explore both options.

As a side note, we did not include content-based recommender in this mix because we would like to leverage the strength of each algorithm. Because content-based recommender yields higher MAE, lower DCG and lower price variances compared to user-user recommender, and results in higher product availability compared to matrix factorization. Also, there are no users here with no ratings (minimum number of rated items is 10 items per user), weakening the usefulness of content-based recommender. Therefore, we included only those two.

The third option is to minimize availability. Because Nile River is good at providing products that consumers could not find easily elsewhere, we would like to explore what could happen if we take recommended products from these three algorithms, and choose top-5 that minimize total availability. In order to explore this option, we should consider all items at once, not considering only products that users have rated. Therefore, the only metrics that we could consider is average price standard deviation and average availability.

3.2 Examples

3.2.1 Equal-Weighted Averages

User 4047

1245 Avery Assorted File Folder Label Pad, 1/3 Cut, 160 Labels (45215)

1240 Post-it Durable IndexTabs, 1 Inch, Ideal For Binders and File Folders, Assorted Bright Colors, 36 per Dispenser (686-RYBT)

1800 Avery Flags, 0.5 Inch, Standard Colors, 100 Flags (22569)

1317 Avery NoteTabs, 2 x 1.5 Inches, Neon Blue and Magenta, 40 per pack (16293)

1324 Avery NoteTabs, 3 x 3.5 Inches, Pastel Blue, 20 per pack (16330)

User 4342

2063 Post-it Full Adhesive Roll, 1 x 400, Pink, 1-Pack ,2650-P

2025 Paper Mate Quick Flip 0.7MM Mechanical Pencil Starter Set, 4 Mechanical Pencils (1808783)

862 3M Mouse Pad with Gel Wrist Rest, Sunrise Design (MW308SR)

1319 Avery PocketTabs, 5.125 x 6 Inches, CD Size, Lime and Blue, 5 per pack (16362)

952 Wilson Jones Big Mouth Filer, Vertical Orientation, Dark Blue (W68583)

User 4462

1298 3M Permanent Adhesive File Folder Labels, 0.66 x 3.437 Inches, White, 1500 per Pack (3300-F)

1342 Avery NoteTabs Round Edge, 2 x 1.5 Inches, Citrus, 20 per pack (16306)

1217 Bankers Box SmoothMove Moving and Storage Boxes, Small, 10 Pack (0062701)

1296 Post-it Super Sticky Removable Color Coding Labels, 1 Inch x 2 5/8-Inch, Assorted Neon, Laser, 450 Labels per Pack (2700-P)

1453 Scotch Restickable Tabs, 1 x 1 Inches, 18 Squares (R100)

3.2.2 Segmented Recommender (< 15: matrix factorization, >=15: user-user recommender)

User 3252 (user-user recommender)

482 Avery Shipping Labels with TrueBlock Technology, Inkjet Printers, 5.5 x 8.5 Inches, White, Pack of 50 (8126)

1296 Post-it Super Sticky Removable Color Coding Labels, 1 Inch x 2 5/8-Inch, Assorted Neon, Laser, 450 Labels per Pack (2700-P)

540 Avery Nonstick Heavy-Duty EZD Reference View 2 Inch White Binder (79192)

1524 Five Star Wirebound Notebook, 5-Subject, 200 College-Ruled Sheets, 11 x 8.5 Inch Sheet Size, Black (72081)

1800 Avery Flags, 0.5 Inch, Standard Colors, 100 Flags (22569)

User 4342 (matrix factorization)

2257 Quartet Push Pins, 1-Inch, Assorted Colors, 30 Pack (27954)

619 PaperPro 1210 Professional 65 Sheet Stapler

327 Wilson Jones Heavy Weight Top-Loading Sheet Protectors, Clear, 100/Box (W21411)

1292 3M Permanent Adhesive Address Labels, 1 x 2.62 Inches, White, 3000 per Pack (3100-B)

1874 Post-it Super Sticky Full Adhesive Notes, 3 x 3-Inches, Assorted Ultra Colors, 4-Pads/Pack

User 4462 (matrix factorization)

2257 Quartet Push Pins, 1-Inch, Assorted Colors, 30 Pack (27954)

619 PaperPro 1210 Professional 65 Sheet Stapler

1292 3M Permanent Adhesive Address Labels, 1 x 2.62 Inches, White, 3000 per Pack (3100-B)

327 Wilson Jones Heavy Weight Top-Loading Sheet Protectors, Clear, 100/Box (W21411)

1297 3M Permanent Adhesive Shipping Labels, 2 x 4 Inches, White, 2500 per Pack (3100-U)

3.2.3 Segmented Recommender (< 15: user-user recommender, >=15: matrix factorization)

User 3252 (matrix factorization)

619 PaperPro 1210 Professional 65 Sheet Stapler

2257 Quartet Push Pins, 1-Inch, Assorted Colors, 30 Pack (27954)

327 Wilson Jones Heavy Weight Top-Loading Sheet Protectors, Clear, 100/Box (W21411)

1874 Post-it Super Sticky Full Adhesive Notes, 3 x 3-Inches, Assorted Ultra Colors, 4-Pads/Pack

2169 Scotch Bi-Directional Filament Tape 8959 Transparent, 50 mm x 50 m, Conveniently Packaged (Pack of 1)

User 4342 (user-user recommender)

2063 Post-it Full Adhesive Roll, 1 x 400, Pink, 1-Pack ,2650-P

257 Scotch Tear-by-Hand Tape, 1.88 Inches x 50 Yards, 2-Pack (3842-2)

1319 Avery PocketTabs, 5.125 x 6 Inches, CD Size, Lime and Blue, 5 per pack (16362)

1328 Avery Shipping Label with Paper Receipt, Laser, TrueBlock Technology, White, 25 Sheets (5327)

1638 Avery Protect and Store View Binder with 1 Inch EZ-Turn Ring, 5.5 x 8.5 Inches , White (23011)

User 4462 (user-user recommender)

1342 Avery NoteTabs Round Edge, 2 x 1.5 Inches, Citrus, 20 per pack (16306)

1453 Scotch Restickable Tabs, 1 x 1 Inches, 18 Squares (R100)

1300 Post-it® Super Sticky Removable File Folder Labels, 0.66 x 3.437 Inches, White, 1500 per Pack (2100-F)

1292 3M Permanent Adhesive Address Labels, 1 x 2.62 Inches, White, 3000 per Pack (3100-B)

1298 3M Permanent Adhesive File Folder Labels, 0.66 x 3.437 Inches, White, 1500 per Pack (3300-F)

3.2.4 Availability Minimizer

User 4047

619 PaperPro 1210 Professional 65 Sheet Stapler

72 Avery Index Maker Clear Label Dividers with 5 White Tabs 25 Count (11446)

129 Five Star Spiral Notebook, College Ruled, 1 Subject, 8.5 x 11 Inches, 100 Sheets, Assorted Colors (06206)

417 Wilson Jones Favorite Desk File/Sorter, A-Z Index, 10 x 12 Inches, Burgundy (WCC3C)

71 Avery® White Shipping Labels for Laser Printers with TrueBlock(TM) Technology, 2 inches x 4 inches, Pack of 250 (5263)

User 4342

619 PaperPro 1210 Professional 65 Sheet Stapler

129 Five Star Spiral Notebook, College Ruled, 1 Subject, 8.5 x 11 Inches, 100 Sheets, Assorted Colors (06206)

71 Avery® White Shipping Labels for Laser Printers with TrueBlock(TM) Technology, 2 inches x 4 inches, Pack of 250 (5263)

1241 Kensington Wireless Presenter with Laser Pointer and 2 GB Built-in Memory (Black)

435 BIC Mark-It Color Collection Permanent Markers, Fine Point, Assorted, 12 Markers

User 4462

619 PaperPro 1210 Professional 65 Sheet Stapler

191 Scotch Filament Tape 893 Clear, 48 mm x 55 m (Pack of 1)

24 3M Scotch Mounting Tape, .5-Inch by 75-Inch (110)

71 Avery® White Shipping Labels for Laser Printers with TrueBlock(TM) Technology, 2 inches x 4 inches, Pack of 250 (5263)

1298 3M Permanent Adhesive File Folder Labels, 0.66 x 3.437 Inches, White, 1500 per Pack (3300-F)

3.3 Summary Statistics

This table shows the evaluation of each hybrid algorithm where MAE and DCG are measured by computing top-5 list from items that users have rated only while Average Price Standard Deviation and Average Availability are measured by computing top-5 list from all items. Note that for availability minimizer, since we computed recommendation from the whole product list, we could output only average price standard deviation and average availability.

	Equal-Weighted Average (EWA)	Segmented Recommender: >= 15 UU, < 15 MF	Segmented Recommender: >= 15 MF, < 15 UU	Availability Minimizer (AM)
1. MAE	0.349	0.445	0.333	-
2. DCG	16.682	16.321	16.756	-
3. Average Price Standard Deviation	14.275	14.504	19.913	19.207
4. Average Availability	0.714	0.590	0.653	0.475

3.4 Evaluation of Algorithms

Consider the first algorithm: equal-weighted average (EWA). Even though this is the simplest algorithm, it outputs product list with low price variance and high availability among these four choices. This algorithm is not recommended.

Consider the second choice: segmented recommender. We could see that if we let users with less than 15 rated items use user-user recommender, and users with at least 15 items use matrix factorization, we could achieve lower MAE, higher DCG and higher product mix in terms of price. Even though this algorithm outputs product with higher availability compared with the other setup, overall, this setup is the best among these two. This also makes sense practically: because user-user recommender could output products with high variety, we could recommend items liked by similar users to that particular user. Also, as matrix factorization could output recommendation based on latent factors, having high enough number of ratings could lead to better results in recommending top-5 product list.

Consider the third choice: availability minimizer. We could see that by minimizing total availability using greedy approach, we could get very low product availability, 0.475. Also, average price standard deviation is still high. This suggests that by minimizing total availability directly, we could still get product mixes with high price variances. These two goals are not in conflict with each other. This also makes sense for this particular case: we would like to sell big and small items together, and we would like to sell products which could not be found easily in other stores. We could achieve these two important goals by choosing this approach.

Part 4: Proposal and Reflection

4.1 Proposal

4.1.1 Business needs and opportunities

The problem is that Nile-River has faced modest increase in sales during promotion back-to-school. Hence, Nile-River has to adopt recommender system that is based on user history profiles. By investing in new recommender system, this could be opportunity for Nile-River also because Nile-River has many competitive edges. For example, Nile-River has deeper product catalogs compared with competitors. In addition, Nile-River has done research that could help guide constructing recommender system. For example, large-dollar purchases result from selling expensive and cheap items together. Driven by the needs to increase sales, Nile-River should be ready to implement the suggested recommender system.

4.1.2 Evaluation criteria

First, we need prediction accuracy metrics as a baseline in predicting how far our prediction is from true ratings. This is captured by MAE. Second, since our goal is to produce top-5 list, we need another measure in capturing how good the whole recommended list is. This is captured by DCG. Third, we need to measure how much we utilized the strength of Nile-River. This is captured by two metrics: average price standard deviation and average availability. Average price standard deviation measures how diverse the recommended list is.

Average availability measures how well we could offer items that consumers could not find elsewhere.

4.1.3 Algorithm Suggestions

First, we considered three areas of recommender algorithms. The first area is non-personalized and content-based recommender. For this one, we chose content-based recommender as we could get some insights from the product itself, such as category and metadata. This is especially useful for new site users. The second area is nearest neighbor recommender. For this one, we chose user-user recommender because user-user recommender finds similar people to that user and recommend items that those people liked to that person. This is particularly useful if we would like our recommender to produce the diversity which is essential for this case. The third area is matrix factorization which captures latent factors that helped predict user's ratings. This is particularly useful if we have enough number of ratings on that user, and would like to predict ratings for other items.

Then, we found that there is no best algorithm in all categories: each has its own strength and weakness. Therefore, we need a hybrid algorithm that combines these three. We recommended two useful algorithms here.

First, segmented recommender where users with less than 15 items rated use user-user recommender while users with at least 15 items use matrix factorization. The motivation is that to add diversity to recommended list, we could find neighbors with similar tastes in order to make large-dollar sells. Also, for users with at least 15 items, we could rely on matrix factorization due to number of data points for that group of users. This could help predicting ratings accurately and adding diversity to the recommender. For example, it could refer to Pocket Tab and Note Tab to the list which is usually full of 3M products.

Second, availability minimizer. The motivation is that this company has deep product catalogs, so by minimizing total availability directly from combined recommended list in base algorithms, we could leverage the company strength. The result shows that this could help reducing availability product a lot while still maintaining diversity of the recommender in terms of price mix. For example, it could recommend Paper Pro Stapler which is not easily available in other stores to many consumer lists.

4.2 Reflection

4.2.1 Data management

Dividing data into train-validation-test split could allow us to measure the usefulness of algorithms directly. The motivation is that we keep the test split (last 20 users) to get an unbiased measure of performance. If we measured performance on train or validation set, and we got good results, we could not say that it is because algorithm is performing well or algorithm is just overfitting to the data. This is useful in both base algorithms and hybrid algorithms.

4.2.2 Reflections

First, we could not only rely on prediction accuracy metrics to evaluate algorithm performances. The key of business goals is not just produce the most accurate recommender, but the recommender that helps people find useful items that they may never have thought of. Therefore, we needed to consider other metrics that match our business goals also, such as diversity and availability.

Second, we need to understand business clearly before jumping to make one recommender algorithm. There is no silver bullet algorithm that works for every business. Because every company has its own strengths, the appropriate recommender is the one that helps leverage business competitive edge with suitable matching evaluation metric. In this case, diversity and product availability are the main concerns for the company.

Third, we could do better than the base algorithms. Each algorithm in each area has its own strengths and weaknesses. In order to combine the strengths of all algorithms to produce effective recommendation, we need hybrid algorithms. By using hybrid algorithms, we could find the sweet spot in segmented recommender that helps satisfying all evaluation metrics. Also, we could leverage company strength more by minimizing product availability from the combined list of three base algorithms.