ABSTRACT
Currently, shopping habits of customers are need based. The general process of buying a product involves considerable research into the products in consideration, followed by a monotonous purchase of the finalized product from an e-commerce website. This research involves watching video reviews, reading customer feedback and ratings, etc. A new way of providing recommendations with increased level of trust and providing a higher level of confidence in discovery of new products to customers is proposed. The second part of this project focuses on recognizing different users accessing a single account to provide more personalized results, research the effects and solutions for the same, and

2. DATA COLLECTION
Data collection for the two recommendation systems, namely, Collaborative-filtering and Social Recommendation Systems, differed on the most fundamental level. In case of the former, it was necessary to get data with product relations based on users i.e. person who bought item A also bought item B. This dataset was item-focused and needed a fair amount of "related purchases" dimension. In case of social recommendation system, data about customers and their trusted friends with similar interest was needed. This data was collected by sending out surveys to RIT students and valuable inputs about these relations. Data collection for user recognition involved creating user profiles based on anonymous surveys from RIT students. These surveys were used to create user profiles with location data, music/video preferences, and different user habits like time of access, playtime for media, etc.

2.1 Collaborative-filtering
Data for this approach was used primarily from Amazon products. The dataset was retrieved from DataDives, which provided a collection of datasets with product categories, titles, brands, reviews, and the most important aspect, related purchases. This data was useful in creating a singular dataset with a substantial amount of products with accurate titles, reviews, and related purchases. Initially, a dataset with 50,000 products was created with related purchases merged in with each product. This dataset was used for testing collaborative-filtering approach. Data entries were increased to 100,000 to observe changes in accuracy and confidence of recommendations.

Table 1 represents the dataset attributes created for this approach. Related purchases were limited to 5. Categories had multiple values for a single product.

2.2 Social Recommendation Approach
Data collection for this system was built on top of the currently available data for former approach. The added dimension of "friends" network and their interests were required for the creation of a sample dataset for social recommendations. This was achieved by sending out a survey to
RIT students on inputs for the same. Following is a list of some of the questions posted in the survey.

1. Choose your interests from the listed categories
2. Specify sub-categories for each category if applicable.
   For e.g. if you selected books, type "fiction", "biography"
3. List user ID of your friends with similar interests
4. (Optional) How satisfied are you with recommendations from digital markets like Amazon, Netflix, etc. Rate between 1 to 10 with 1 being low degree of satisfaction and 10 being the highest

It was important to avoid duplicate surveys from the same set of friends. A unique survey ID was allotted to each user and a unique ID of this user’s friends.

The data from the survey was used to create mock users with categories selected by survey-takers used to create sample product affiliations with the user profile.

Table 2 represents the dataset features for each user in Social Recommendation approach.

### 2.3 Multiple user recognition

The data required for this approach involved collecting various personal profile features like location, play times of media, shopping habits centered around different days of a month, genres of items bought/consumed. This called for anonymous surveys to provide privacy in accordance with ethical standards. Anonymous surveys were sent out to roughly 40 students to create user profiles based on their replies. It was required to create their usage history which was possible through different variations and original inputs from these surveys. A usage history over the course of 40 days was created as input to the machine learning system.

The initial approach focused on digital markets similar to Amazon. Over the course of project, due to fuzzy data in this domain, the focus was changed to digital markets like Spotify where the data was found to be more personal and helped in making a more focused research subject than delving into various markets.

It was important to allow the system to learn user habits over a considerable course of time. To simulate this, these 40 user profiles were maintained by running scheduled queries to the data collection system with user habits for the respective profiles. This was a better approach than creating the entire dataset and feeding it into the system as a whole. It also allowed to test the system more frequently and record intermediate results.

### 3. APPROACH

#### 3.1 Collaborative-filtering

This approach was implemented based on item-based collaborative filtering algorithm. This algorithm can be used by selecting one common feature between two items and using this feature to calculate the similarity between the two items. By calculating these similarities, we can design a recommender system to predict values for a customer and item relation without a rating.

The basic idea behind this algorithm is to find customers that have rated both items in consideration. Now the ratings given by customers to both these items are analyzed to calculate the degree of similarity between the two items and recommend one for the other. There are a number of ways this similarity can be calculated. For this implementation, cosine-based similarity was used.[1]

Cosine-based similarity uses the ratings provided for each item and converts them into vectors. The angle between these vectors is calculated to find out similarity between the two. This similarity-based approach is used to create a model for predicting the rating a user might give to the recommended item, meaning, a sense of how much the user might like the recommended product.

Once we have established how the model is created, we store a number of similar items for each product. These similar items are retrieved in response to the product and every similar item that might have been rated by current user is shortlisted. Now, the similarity between these shortlisted items that user has rated and target item is calculated. Based on this calculation, we can predict the rating that user might have given to the recommended product.

#### 3.2 Social Recommendation Engine

The first step in implementing this system was to create user graphs based on their immediate friends and depth of transitive friends selected by the user. This user graph will be used by the recommendation engine to classify which users provide the most useful recommendations to the user. It will also record the level of usefulness of transitive friends and provide user with data so as to add that friend to his list of immediate friends.

The market for this approach was kept limited to the database of Amazon products created for the Collaborative-filtering approach. This helped in comparing the two in a common market and observe the effects for the same. This approach used real-time data over the course of 40 days from 40 users accessing the system on a daily basis to rate recommendations and provide a mock browsing and purchase history for sample products. New products (on suggestions of users) were added at timed intervals to test how a product without any reviews or ratings history would be accepted.

### Table 1: Data for Collaborative filtering system

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>productId</td>
<td>Numerical - Product ID</td>
</tr>
<tr>
<td>product_title</td>
<td>Text - Product Name</td>
</tr>
<tr>
<td>categories</td>
<td>Text - Multiple categories for a single product</td>
</tr>
<tr>
<td>review_score</td>
<td>Numerical - Average review score</td>
</tr>
<tr>
<td>review_userId</td>
<td>Numerical - user ID of reviewer</td>
</tr>
<tr>
<td>also_purchased1</td>
<td>Numerical - Product ID of related purchase</td>
</tr>
<tr>
<td>also_purchased2</td>
<td>Numerical</td>
</tr>
<tr>
<td>...</td>
<td>Related purchases limited to 5</td>
</tr>
</tbody>
</table>

### Table 2: Data for Social Recommendation system

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>userID</td>
<td>Numerical - User ID</td>
</tr>
<tr>
<td>product_ID(s)</td>
<td>Numerical - Shopping history</td>
</tr>
<tr>
<td>friends_list</td>
<td>Numerical - list of friends</td>
</tr>
<tr>
<td>transitive_level</td>
<td>Numerical - Level of transitive friends selected by user</td>
</tr>
</tbody>
</table>

[1] Cosine-based similarity uses the ratings provided for each item and converts them into vectors. The angle between these vectors is calculated to find out similarity between the two items. This similarity-based approach is used to create a model for predicting the rating a user might give to the recommended item, meaning, a sense of how much the user might like the recommended product.
into the market. This helped in observing effects on the cold-start problem that plagues the current system.

The user’s browsing history was recorded every day and input to the evaluator which was responsible for calculating the affected browsing history, purchase history, and discovery of new products. These three performance coefficients were used to rank the users’ friends and update the user graph over the testing period.

The change in browsing history was calculated by recording user habits. Any deviation from the visited products was recorded as a positive reflection on recommendations provided by the system. The system also tracked the genres of products that user was interested in and marked new genres visited by the user during browsing time.

The change in purchase history was relatively straightforward to track with mock purchases made by the user during the testing period. The purchases made through recommendations produced by the products counted towards the rank of the friend who was responsible for the recommendation. Purchase history was given slightly more importance than affected browsing history.

Discovery quotient was an important measure of the usefulness of the system. This lets us evaluate how the system can expand the users’ likes and encourage exploring the digital market for unknown genres and products. This change was tracked by calculating the change in “favorite genres” of the users by introduction of new products as mentioned earlier. As these products were recommended by the users themselves, the users with these friends in their user graph picked up the product in their recommendations.

3.3 Multiple User Recognition

The model for multiple user recognition was created using the various attributes available to the system through anonymous surveys from RIT students. These attributes were used as features for pattern generation for each user. An interesting study was found [3], where a user-dependent model could be created based on time-dependent data and activity logging. This model helped us generate activity graphs for each user and determine the difference between these graphs over the testing period to find anomalies in the patterns and thus, recognize secondary users. Using this graph, we were able to observe how features related to each user affected the user-recognition algorithm. The model uses Expectation Maximization to generate maximum accuracy in estimation of which user is accessing the account currently. We were able to create around 43 features mapped to time and activity data that were used to generate the model for each user.

The focus of this research was for a market like Spotify. Other markets like Amazon and Netflix were also considered for this system. Using a system like Spotify provided us insight into more personal systems when compared to Amazon and Netflix. We were able to use contextual data like location of the user, map play times to activities, and learn about favorite genres of the user depending on the time of the day/week.

Over the course of 40 days, 40 user profiles were maintained by simulating user habits recorded through the surveys and graph generation for the same. This simulated activity logs were then fed to the machine learning system for generation of graph deviation for secondary user activity generation. User profiles were overlapped every 2 days to observe the effects of anomalous activity in the user graphs. Fig. 2 shows the activity generation of a single user for time-dependent data features. As these 40 user profiles were simulated and catered to over the course of 40 days, we were also able to use human intuition to rate some of these recommendations that were classified as not satisfactory by the system. This helped visualize where the system was faltering and how many useful recommendations it was classifying as unsatisfactory. The number of users for a single account were kept limited to 2.

Figure 1: System Design for Social Recommendation System: A-Data store, B-Recommendation engine, C-Activity log, D-Evaluator

4. RESULTS

4.1 Collaborative filtering approach

Initially, 50000 products were compiled into the dataset for building the recommendation system. It was necessary to use existing users for testing recommendations as new users have the cold-start problem. This is attributed to the new users not having any reviews on the system which is required for our similarity prediction.

The quality of recommendations were evaluated, initially, based on two factors, namely, degree of confidence and intuition. Degree of confidence was based on degree of similarity between the target product and the recommended product. This similarity calculation is different than what
we use to calculate the actual recommendation. This similarity calculation is done to evaluate how useful the recommendation was to the user. This also involves the second factor of evaluation which is intuition. In this approach, recommendations were checked manually and evaluated based on personal opinion.

Degree of confidence for recommendations against 20000 products was found to be 68% on an average. By intuition, more than 60% of recommendations were found to be viable. Although, this system does not represent the current systems in markets as accurately as possible, it is helpful in studying the pitfalls and challenges behind implementing these systems.

This approach was useful and pretty accurate when ratings and amount of users used to evaluate the similarity functions were kept limited. This was done to reduce complexity of the algorithm and turn down resource and time requirements to create recommendation system model. This also denotes the problems associated with scaling in this approach. As the number of products increases, the amount of similar products for a target item increases. This means, the amount of similarity calculation and the time required for the same increases exponentially if the comparisons are not contained. To counter this, a limit of 40 products was executed for comparison of similar products.

It is necessary for new products and new users to have ratings and reviews to be useful in the recommendation system. This problem as mentioned before is called “cold-start”. These new products and new users take a longer time to become available for recommendations.

Another important factor was to consider the sparsity of data available on users as well as products.[4] It is important to have data on various types of users and products to provide viable recommendations for every type of user. Over the course of testing, it was found that recommendations tend to skew in one direction meaning one kind of opinion (rating) is always overpowering the others.

Addition of new products proved to be a pitfall for this system. These new products had no review or rating history. This introduced a problem known as cold-start as mentioned before. The product having no history made it difficult to be included in any sort of recommendations unless a considerable number of users rated the product. Out of 40 new products that were introduced into the system during the testing period, only 8% of the same were picked up by the system. The reason for this inclusion was attributed to the fact that they were very similar to or add-on products for existing products which had a low number of similar existing products.

4.2 Social Recommendations approach

The evaluation of Social Recommendations system was based purely around the calculation of performance coefficients, namely, affected browsing history, affected purchase history, and discovery quotient for introduction of new products. The users had the option of rating these recommendations on their scale of usefulness, add it to their favorites, and subscribe to the new genre of products in recommendations. These coefficients were normalized to scale of 1-10 and were used to form a heuristic measurement of usefulness related to recommendations produced by users’ friends. Fig.3 shows a visualization of user graph at the end of testing period with a transitive friends depth of 4. The blue figures denote high ranking of friends and useful recommendations from the same, whereas, the dark figures denote friends whose recommendations are being rated low by the user.

The browsing history changes were tracked by logging all user activities when access the application provided for testing. At the end of their browsing session, these logs were used to create a browsing review which was compared to their existing browsing patterns and all changes were recorded. The browsing patterns for each user are stored in the form of decision trees updated every browsing session. These changes were then measured in terms of deviation from genres attributed to the user, and their rating for these deviating recommendations. If the recommendations were marked as useful by the user, the corresponding friend responsible for the recommendation was given a score based on the rating given by user. The changes in browsing patterns of users’ was observed to be on par with the collaborative filtering approach. But the onset of these changes was observed to be quicker in the social recommendation approach. This means that the system was able to create useful recommendations for the user with less set up time. The user graphs were useful in creating recommendations in the users’ areas of interest. Over the course of 40 days testing period, users started receiving useful recommendations as soon as the 5 days mark as opposed to the collaborative filtering approach making useful recommendations by 14 day mark. This shows that the social aspect of the system introduces a level of control for the user to tailor their recommendations. This provides a significant advantage to the system as opposed to the collaborative filtering approach where the system is responsible for the entirety of tailoring product suggestions for the user. As we observe in Fig 4., a normalized graph of accuracy against time period, usefulness of recommendations spiked earlier than that of collaborative filtering approach.

Users were given the option to make mock purchases of products to simulate how likely was the user to buy products recommended by the system. Purchases were given an increased score as opposed to browsing history by a factor of 2. These purchases were also helpful in providing a look into the usefulness of the recommendations when compared to the collaborative filtering approach in the way that the products in the latter approach provided a level of trust higher than the former. This can be attributed to receiving a recommendation from a friend that you trust. The time between recommendation and purchase of the product was significantly lower, by 92%, in the social recommendation approach. Having said that, the existing systems provide a much more useful system based on user purchases. Current systems are able to produce accurate recommendations of products related/add-on for the product that the user just purchased. The purchase history was observed to be much more profitable when compare to the social recommendations approach. Looking at Fig. 5, although initially on par, the collaborative filtering search produced better results related to purchases as it provided more contextual products to the user based on their purchase history.

The discovery quotient of new products was the last coefficient on which the friends’ usefulness was rated. This is not the same as solving the cold-start problem. Discovery quotient talks about introduction of new genres to the users that they were unaware of, or introduction of new products in their area of interests that was unknown before. A signif-
significant increase in introduction of new products was observed with the social recommendation approach. Over 40 users, about 90% of users found new genres they were interested in, and about 80% of users received recommendations of new products in their area of interest. These users are those that either made the mock purchase of these newly discovered products or added them to their wishlist (favorites). Approximately 60% increase was observed in browsing and purchasing habits of users related to new products as opposed to a measly 12% increase in the current systems. Looking at Fig. 6, we observe a considerable increase in recommendations involving new products as they have the confidence of the friend that is responsible for the recommendation. A one-to-one mapping is created to the confidence of the product without any setup time (reviews and ratings) as opposed to the case in collaborative filtering approach.

The social recommendation approach was especially useful in introduction of new products to the system. As mentioned earlier, new products were introduced into the system every 2 days during the testing period. As these products were recommendations from the users themselves, they received a higher level of confidence as opposed to new products in the collaborative filtering approach. Also, as these recommendations travel along user graphs and can affect recommendations for other users too, the new products were able to gain a strong foothold in the market with minimal time. On an average, out of the 20 new products introduced every day by 10 different users, close to 81% of the products were well received by users and review history for the same was created in a significantly shorter time.

4.3 Multiple-user recognition

For each user, the data over the 40 day testing period was split into 66% for training and 33% for testing. The recommendations generated from the training data were used to match the activity logs in the testing data. Similarity between the recommendations and the testing data was calculated using cosine-based similarity as mentioned in the collaborative filtering approach. Using this calculated similarity, we were able to predict how satisfied the user will be the recommendations. This data was subjected to an evaluation algorithm to compare accuracy between different digital markets - Amazon, Netflix, Spotify.

For the simulation of Amazon and Netflix market, user activities were mapped to time-dependent data from user profiles. This approach created satisfactory results in terms of recognizing the original user for a user profile. In case of recognizing secondary user, the system generated a high ratio of false positives where the primary user was classified as secondary user. This was attributed to the fact that these markets were generally used in social aspects as compared to personal use. The time spent by the original user for personal use was very similar to the time spent by the user for social reasons. The activity data was especially disparate for Amazon market data as the user interests changed widely and also catered to shopping for friends, relatives, different occasions throughout the testing period. Looking at Fig. 8, we observed slightly better performance in secondary user recognition for the Netflix environment. Out of 2,000 recommendations created over the course of the testing period, about 58% accuracy was noted in recognizing secondary user and filtering out suggestions for the primary user. In case of Amazon, we observed a sharp decrease in the accuracy of the system recommendations over the course of time as different events and usage habits of user affected the recommendations. We observed out of the 2000 recommendations generated, about 50% of valid original user activity was ignored due to disparity between the original user patterns created and the updated patterns.

In case of Spotify market, we observed a 10% increase in accuracy of the original user recognition over other markets. This market was able to provide us with more contextual
data as it was being used by users almost everyday with little random usages. The time-dependent data attributed to more 70% of accurate recommendations. In case of recognizing the secondary user, we saw steady increase in the accuracy of the recommender system. We observed the system was able to estimate a secondary user with an accuracy of 70% on an average. This denotes an approximate 25% increase in accuracy over other markets.

It is important to note that users more frequently accessed a market like Spotify over a market like Amazon. This provides more contextual information and reduces the sparsity of data. It also allowed the system to learn user habits more accurately and classify less valid recommendations as unsatisfactory. The false positives in case of Spotify environment reduced by almost 80% when compared to Amazon and Netflix simulations.

The main issues in implementing item-based collaborative filtering was to keep the parameters of the algorithm contained. The algorithm’s complexity increases exponentially as data is scaled up. Even with data records kept in limit, the amount of comparisons for calculation of similarity needs to be limited. If the amount of similar items compared to calculate recommendations is not limited, the amount of time and resources required increases exponentially.

The amount of users with reviews for a common product should be limited as well. Sometimes, there are not enough amount of users to calculate recommendations with an acceptable level of confidence and/or similarity between users. This causes the recommendations to be vague and are not helpful to the user.

For the user recognition aspect of the project, it was very difficult to expect testers to provide varying parameters like consistent play times, personal information like varying locations, and creating scenarios where multiple users would use the same account. An initial set of surveys were sent out to 20 RIT students so as to create initial user profiles. These profiles were then maintained by simulating parameters from surveys over the course of a 20 day testing period. This required significant effort as the system needed to be updated with data everyday with different play times of genres of songs, and differing locations.

The initial approach of user recognition focused on digital markets like Amazon. This proved to be a difficult market to implement user recognition as the user catered to differing purchasing habits based on holidays, events like birthdays, etc. For example, a male user may be browsing gifts for his sister’s birthday and this would record as a different user by the system as it deviated from the normal shopping habits of the user. The false positives were too high in this digital market. The system was able to recognize a different user with an accuracy of 63%, whereas, it wrongly classified the original user as a different user with an accuracy of 48%. This experiment needed a more personal digital market that the user would use often, ideally every day. This is why the focus was changed to cater to a market like Spotify.

5.2 Collaborative filtering approach

The main issues in implementing item-based collaborative filtering was to keep the parameters of the algorithm contained. The algorithm’s complexity increases exponentially as data is scaled up. Even with data records kept in limit, the amount of comparisons for calculation of similarity needs to be limited. If the amount of similar items compared to calculate recommendations is not limited, the amount of time and resources required increases exponentially.

The amount of users with reviews for a common product should be limited as well. Sometimes, there are not enough amount of users to calculate recommendations with an acceptable level of confidence and/or similarity between users. This causes the recommendations to be vague and are not helpful to the user.

5.3 Social recommendation approach

The initial approach included research into how many transitive friends are enough to provide the user useful recommendations. This involved a deeper study into how satisfied the user is with recommendations and how much is enough. This would also involve studying how complex the user graphs becomes where updating and generating recommendations is a task that is detrimental to the recommendation engine. This would require investment into algorithm complexity studies, a more involved testing method which consequently would require more participation from the testers. It was a difficult task to get accurate readings on the recommendations generated from testers. As an alternative to this, the control over transitive friends was given to the user. As the algorithm was based on iterative deepening search, the user could specify the level of transitive friends and the search would be limited to that depth when generating recommendations. Although, with this solution, it could be possible that the user would miss out on valuable recommendations (high information gain) from transitive friends. To counter this, the system was designed to provide the user...
with suggestions to add these high ranking friends (nodes) to their immediate friends list. The user was kept in control of the friends they want to follow.

It was important to limit the number of products introduced in the system as the number of users were relatively less. The original dataset included over 100,000 products and this created a lot of noisy data in the recommendation engine as only 20% of the data was being used for user recommendations. The dataset was reduced using feedback from testers about their areas of interests and the dataset was reduced to approximately 30,000 products with enough variation to cover the different interests of users.

5.4 User recognition approach

The initial approach for this system used the Amazon market data to create user profiles for over 40 different users. It was apparent after initial tests that the Amazon environment was less personal when compared to systems like Spotify and Netflix. It lacked much contextual information and the products covered everything from personal products like cosmetics to camping gear for outdoors. There was much sparsity in the data and even more sparsity was found when surveys were completed by testers. Their interests varied greatly and it was difficult to keep track of these many factors in the user recognition engine. The features for each user were found to be very limited whereas the features itself had over 10,000 different values for each. For this reason, the market was changed to Spotify where we were able to limit the data and focus on a more personalized system. We were able to leverage location and time dependent data which was relatively sparse and varied greatly in the Amazon market environment.

6. CONCLUSION

Current systems provide a really good recommendation systems focused around products. Although, collaborative filtering approach finds similar users related to you, this is done based around the specific products that you are interested in. This limits the customization that user could achieve with a social recommendation system. The proposed system produced useful recommendations in 15% less time than the existing systems. This provides the proof that social recommendations are useful in starting off recommendations for the user at a much faster pace, and consequently study user habits related to the same and design a usage pattern much accurately with less time. This also touches on the cold-start problem that currently plagues the existing systems where a new product takes a considerable amount of time before gaining a history from reviews and ratings. The proposed system drastically reduced this time by 90% as friends in user networks suggest this product with an increased sense of confidence.

Although, the proposed system performs well in the scenarios mentioned above, it lacks in providing contextual recommendations related to products. For example, if a user buys a new laptop, collaborative filtering system would provide immediate recommendations like accessories for the product the user just bought. Thus, we believe, the proposed system may not replace the current system but would definitely provide great results if used in conjunction. The proposed system also aims to resolve issues like recommendation confidence where the user is involved in extensive research before making a purchase whereas a recommendation from a friend may no be subject to the same critic.

We believe the proposed system puts the user in control of their recommendations and creates a sense of trust based around the same. This is an important aspect when it comes to any digital market and provides an insightful look into the social aspects of e-commerce.

In case of multiple user recognition in digital markets, the proposed system provides insight into what kind of data is required to make an approximate estimation of the current user. The proposed system produced successful results in case of more personalized markets like Spotify where more contextual information was available.

7. REFERENCES