

# Final Report

## Team 51

John Kiley, Brian Merrill,  
Hemant Patel, Julia Rodd



# Table of Contents

<b>The Opportunity</b>	<b>2</b>
About Us	2
About Our Target Company	2
Problem Statement	3
Solution	3
Business Benefits	4
Project Goals	5
Business Questions	5
Deliverables	6
Project Status	6
 <b>Approach</b>	 <b>6</b>
Technology	6
Data Overview	7
Recommendation Engine Methods	9
Modeling Approach	12
 <b>Discovery and Results</b>	 <b>18</b>
Data Preparation	18
Reviewer Features	18
Product Features	20
Review Text Features	23
Results	24
Baseline Model	25
Background	25
Phase 1 Build	26
Phase 2 Build	27
Review Text Model	29
Phase 1 Build	29
Phase 2 Build	32
Deep Learning Model	32
Background	32
Phase 1 Build	33
Phase 2 Build	36
Ensemble Model	42
Dashboard and Mobile Application	43
 <b>Conclusions and Recommendations</b>	 <b>47</b>
Conclusions	47
Phase 1 Recommendations	49
Phase 2 Recommendations	51
 <b>References</b>	 <b>54</b>
<b>Appendix</b>	<b>56</b>
Code	56
Dashboard and Mobile Application	56
Project Team	57
Table 1 - Electronics Data Overview	58
Table 2 - Reviewer Features	60
Table 3 - Product Features	61
Table 4 - Baseline Model Cross Validation Results	63
Figures 1 and 2 - Reviewer Text Clusters	63

# The Opportunity

## About Us

We at CognoClick know what it takes to build a successful recommendation engine. We have worked with clients across a wide array of industries and have helped each one exceed their sales goals through effective product recommendations. Using artificial intelligence and state-of-the-art natural language processing (NLP) techniques, we have helped clients make better use of their purchase history and product review data. Our proprietary solution allows clients to retain existing customers and attract new ones by offering highly relevant products.

---

*"Our cognitive and predictive analytics will increase customer sales by identifying high relevance products for both new and existing customers."*

---

## About Our Target Company

Amazon.com Inc. (Amazon) is a worldwide e-commerce leader and technology company that offers various products and services, ranging from online shopping to video streaming and web services. With a mission "to be the Earth's most customer-centric company," Amazon has experienced great success (Amazon's global career site). In 2018, Amazon reported over 141 billion dollars in product sales and over 232 billion dollars in earnings across U.S. and international markets (AMZN.O - Amazon.com, Inc. Profile). Looking beyond the numbers, Amazon has had a lasting impact on the online shopping experience. By carrying millions of products across a multitude of categories, Amazon is able to appeal to a wide variety of customers and distinguish itself from the competition through customer choice.

## Problem Statement

Since its inception nearly 25 years ago, Amazon has been focused on delivering a world class customer experience. Today, in a world where consumer choice is paramount, Amazon continues to need better ways to target new and existing customers by presenting them with products they are most likely to purchase. The key to solving this problem lies in one of Amazon's greatest data assets: **customer reviews**.

Customer reviews have been the cornerstone to Amazon's success. According to Vega (2017), BloomReach, a marketing research firm, found that "Amazon product reviews are the most popular and trusted, and that customers will go to Amazon for reviews even if they intend to purchase the product elsewhere." Understanding the relationship between customer reviews and purchase decisions becomes essential in positioning Amazon for continued growth.

Amazon has a subset of customers who write reviews about their purchased products (reviewers). Reviewers tend to be more engaged with Amazon's platform, giving Amazon the ability to increase product exposure with this segment. By mining historical reviews for information on product preference, Amazon can develop a more personalized product recommendation experience for each reviewer.

## Solution

CognoClick will expedite Amazon's opportunity to drive business value by creating a more personalized product recommendation experience for its reviewers.

CognoClick will implement a 10-week proof-of-concept (POC) for a revamped recommendation engine by leveraging data from the electronics product category. By starting small, CognoClick will gain the incremental feedback needed to prepare for scaling across Amazon's other product categories. In taking this approach, Amazon will better understand the roles and responsibilities needed to support the recommendation engine effectively.

---

*"You have to invent for customers, because companies that only listen to customers fail. It's not the customers' job to invent for themselves. It's our job at Amazon.com to invent...those kinds of things that customers really like" (Jeff Bezos, CEO of Amazon, as cited in Economy, 2019).*

---

Amazon's enterprise dedication to the customer positions itself as an ideal partner for analytics as a service. In addition to Amazon's core values, Amazon also boasts fully integrated marketing, operations, and analytics teams that are well positioned to operationalize a tool built off of customer reviews.

To increase understanding of the revamped recommendations, CognoClick will provide Amazon's marketing, operations, and analytics teams with an interactive dashboard and mobile application.

## Business Benefits

While Amazon's business opportunity is clear, the benefits of a recommendation engine are not to be overlooked. This information helps to provide justification into the investment of a revamped recommendation engine.

The increase in relevant content to consumers has been demonstrated to significantly **increase revenue**. The science and art of automatically recommending content to customers has been the hallmark and competitive advantage of many online service providers. For instance, Netflix's recommendation engine is so critical to its product differentiation that the company valued it at over one billion dollars per year (McAlone, 2016).

In addition to bottom line impact, recommendation engines offer softer benefits to organizations that are willing to implement them. Kordík (2016) found that recommendation engines provide organizations with **better insight into the needs of their customers** and the types of the products being sold. Learning customer preferences allows for the personalization of e-mail and other intelligence-directed marketing to increase the likelihood of return visits. Moreover, these insights can be transformed into key performance indicators (KPIs) that an organization can then use to determine where to invest, which products to source, and which markets to consider for expansion (Kordík, 2016). Figure 1 summarizes the benefits of recommendation engines, ranging from increased revenue and user engagement to overall cost savings through automation.

Automated Recommendation Benefits	
Revenue	5% - 15% overall revenue increase
Engagement	12 - 18% of visitors are engage with product recommendations
Average Order Value	30% - 70% increase - visitors who engage with recommendations
Items Per Order	20% - 40% increase - visitors who engage with recommendations
Conversion Rate	2 - 4x increase - visitors who engage with recommendations
Save Staff Time	Elimination of manual content management efforts

Online Marketing and Business Optimization

CORE METRICS

Figure 1: Recommendation Engine Benefits (“Automated Product Recommendations”, 2009)

One of the immediate effects of sharing more personalized content with users is **increased engagement**. For example, AVARI, an automated campaign personalization service, experienced a significant increase in click-through rates when customers were emailed personalized content (Hathaway, 2015). Figure 2 captures AVARI's lift results, which helps to demonstrate the positive impact that personalization can bring to organizations.

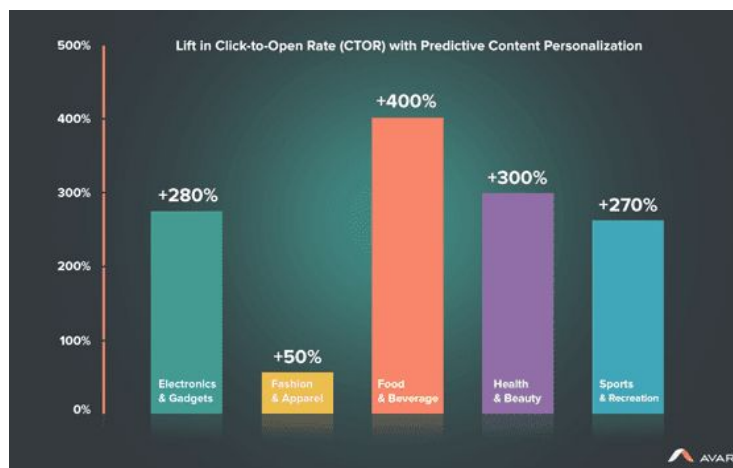


Figure 2: AVARI Email Campaign Results (Hathaway, 2015)

By delivering personalized and high relevance product recommendations to reviewers, CognoClick will help Amazon experience the many benefits of a recommendation engine, including increased product impressions and better insights into the needs of a key customer segment. In making this investment and partnering with CognoClick, Amazon will experience an immediate revenue lift in the short term and significant return on investment over the long run.

## Project Goals

CognoClick's project goals and business questions for the recommendation engine POC within Amazon's electronics product category were streamlined a few weeks into development. The project goals are as follows:

- Assess different modeling approaches and build a recommendation engine that best recommends products to each reviewer
- Build a recommendation engine that incorporates both review text and metadata features
- Leverage NLP techniques to process review data and incorporate keywords as well as product and user features into the model
- Build and implement a model framework that is generalizable to Amazon's other product categories
- Create an interactive dashboard and mobile application that allows Amazon to monitor results and business impact
- Define additional opportunities for the current recommendation engine that will be addressed in a Phase 2 implementation

## Business Questions

By delivering on the above mentioned project goals, CognoClick will answer the following questions:

- Which product(s) are most similar to one another?
- Which product sub-categories and products are most relevant to each reviewer?
- Are product or user recommendations more effective?
- Which product and marketing opportunities should Amazon consider for future investment?

## Deliverables

CognoClick has four main deliverables for the POC:

1. Recommendation engine built in Python 3.7.x
2. Interactive dashboard and mobile application for Amazon's marketing, operations, and analytics teams
3. Report describing the modeling approach, solution, and business impact
4. CEO presentation summarizing project goals, results, and dashboard/mobile application

## Project Status

The CognoClick team is on track to complete the 10-week POC. The team continues to remain in close communication and has been committed to delivering above and beyond project goals. The only major remaining deliverable is the CEO presentation, and there are no risk items to report. Figure 3 recaps the project timeline and status.



Figure 3: Project Timeline and Status

# Approach

## Technology

CogoClick utilized the following technologies to deliver on project goals and complete all deliverables:

- Python 3.7.x (latest available at the time of the POC) for data exploration, visualization, data preparation, and modeling.
- Tensorflow 2.0 with Keras 2.3.0 (latest available at the time of the POC) for building deep learning models.
- Git and Github for code management and collaboration. Git LFS (Large File Storage) was also set up to support collaboration with the large data sets used in this POC.
- Google Cloud to help scale the data preparation and model-building work and enable faster speed-of-delivery.
- Dash, a Python framework for building web applications, for creating the dashboard and mobile application.
- Heroku, a cloud platform for deploying and managing applications, for hosting the dashboard and mobile application.
- Google Drive for document management and storage.

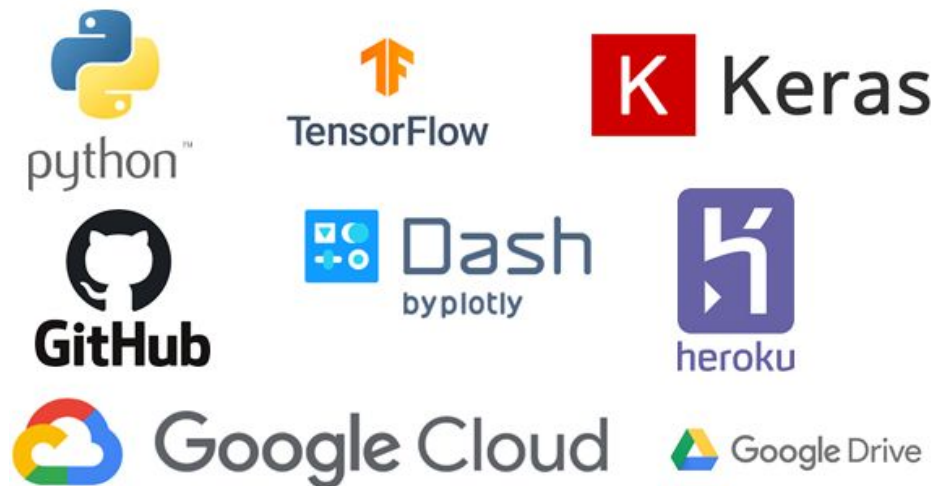


Figure 4: Technology Overview

## Data Overview

The POC is using Amazon product reviews data, which was released by Amazon for research purposes. The data contains over 130 million customer reviews from 1996 to 2014 across multiple product categories.

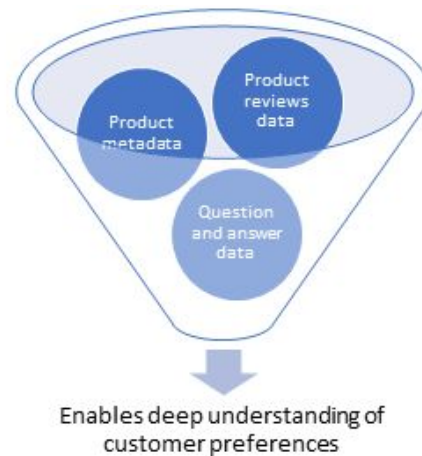
In researching how best to source this data, the CognoClick team discovered neatly cleansed and packaged datasets created by Julian McAuley, a professor at UCSD. More importantly, Julian McAuley also packaged other relevant product metadata to help inform the reviews data, eliminating the need for web scraping.



Julian McAuley has done significant research with recommender systems and is a trustworthy source to use for this project. He has spoken on this topic at multiple conferences, and therefore, the CognoClick team is utilizing his curated data files in order to deliver on project goals (McAuley, n.d.). All data files are in a json format.

More specifically, the CognoClick team is using three types of data within the electronics product category to build the recommendation engine. Table 1 in the Appendix provides examples for the fields, definitions, and sample values for all three data sets.

1. **Product reviews data**, which contains the review text and product ratings given by a reviewer.
2. **Product metadata**, which contains product-specific details, such as product category, price, and sales rank for each product.
3. **Question and answer data**, which contains the question and answer text for a product.



The electronics **product reviews data** contains reviews from June of 1999 through July of 2014. In total, there are approximately 1.6 million reviews that have been completed by over 192 thousand reviewers across 63 thousand products. The reviews data contains nine columns and has been scrubbed to only include products and reviewers that have left at least five reviews or more. Figure 5 shows the distribution of reviews by product rating. Most of the reviews in the electronics data set (81%) have a positive rating (4.0 or 5.0).

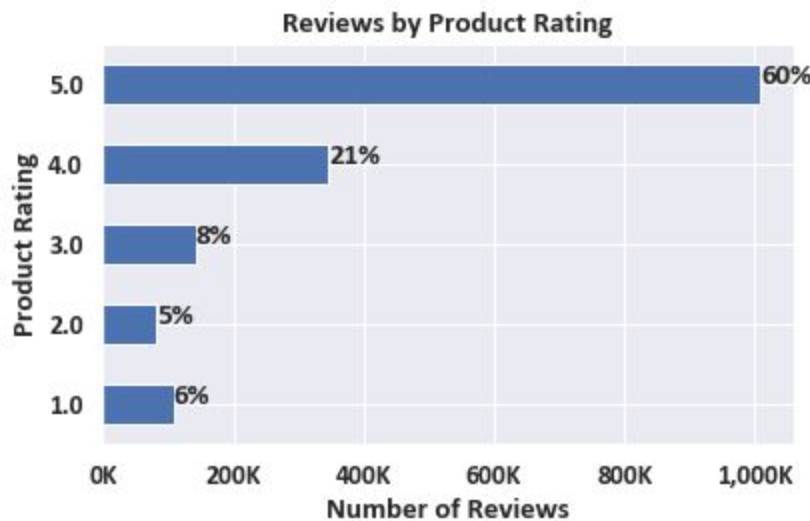


Figure 5: Distribution of Reviews by Product Rating

The **product metadata** is comprised of just over 498 thousand products and contains eight columns in total. The product metadata was captured at a single point in time circa August 2014.



Upon further inspection of the product metadata, it was discovered that several products are not considered to be electronics. When joining to the product reviews data, it was revealed that 11 products deviate from the electronics category (see Figure 6 below). As a result, these 11 products from the 'Automotive' and 'Clothing, Shoes & Jewelry' categories were removed. This filtering resulted in a reduction of 413 reviews but did not reduce the number of reviewers considered.

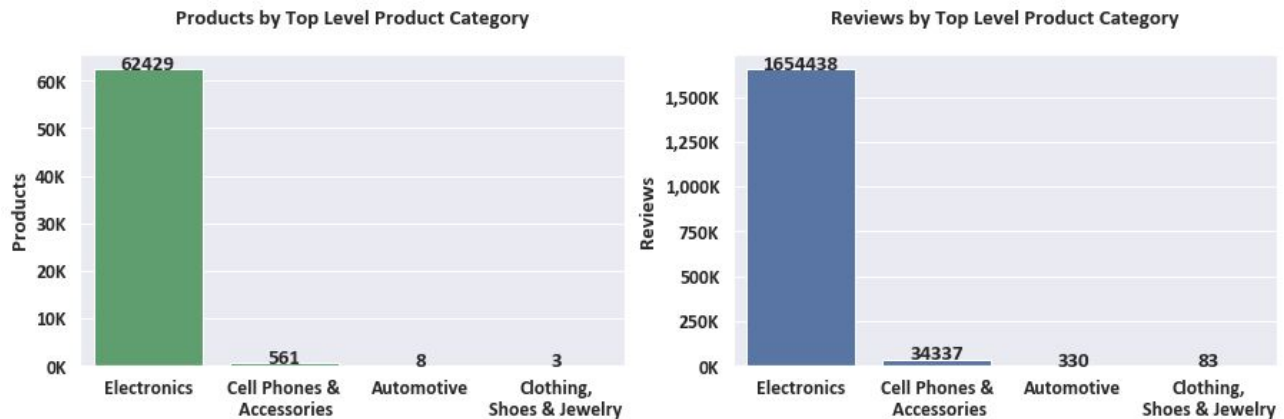


Figure 6: Products and Reviews by Top Level Product Category

Lastly, the **question and answer data** contains approximately 314 thousand rows of data and seven columns. After filtering the data to the relevant electronics category, only 23 thousand products have a question or answer (37% of all products).

By utilizing the three data sets mentioned above, the CognoClick team will be able to leverage both review text and metadata features to gain a full picture of reviewer preferences.

## Recommendation Engine Methods

The CognoClick team performed extensive research to determine which method is most appropriate for building Amazon's revamped electronics recommendation engine. This section provides background on some common recommendation engine methods, which ultimately sets the stage for the direction that the CognoClick team is taking for the POC.

---

*"The basic task of a recommender system is to suggest relevant items to users, based on their opinions, context, and behavior" (McAuley et. al, 2015)*

---

There are many different methods and algorithms that can be employed to recommend products to users. Overall, there are three industry-standard methods:

1. **Content-based filtering method**, which recommends items to a user based on historical reviews or interactions by that user.
2. **Collaborative filtering method**, which recommends items to a user based on historical reviews or interactions by other similar users/items.

3. **Hybrid method**, which recommends items to a user by combining results from both content-based and collaborative filtering methods.

An overview of the recommendation engine ecosystem can be found in Figure 7 below. Each method will be described in additional detail.

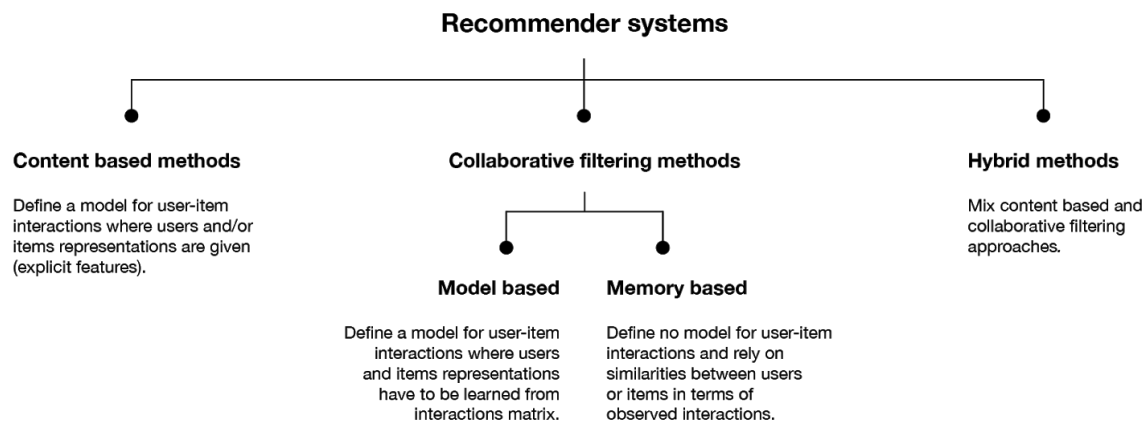


Figure 7: Recommender systems methods (Rocca, 2019)

The **content-based filtering method** examines the details of an item, such as its rating, title, description, and price, and attempts to find similar or replacement items that are most like the identified item. One of the defining attributes of content-based filtering is that only a user's past behavior is considered. Therefore, with this approach, a recommendation engine is constrained to only recommend items that a user has liked or bought in the past (Sharma, 2019). An illustrative depiction of content-based filtering as well as collaborative filtering is shown in Figure 8.

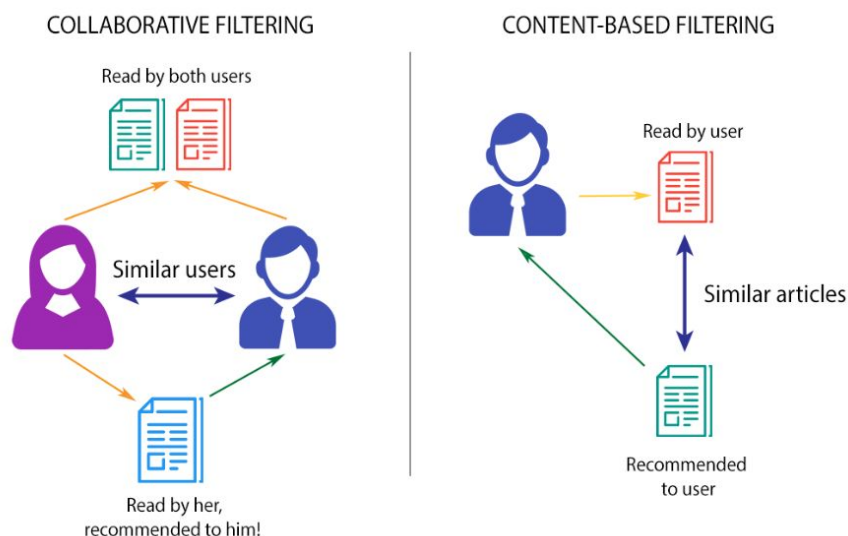


Figure 8: Collaborative and content-based filtering (Liao, 2018)

On the other hand, the **collaborative filtering method** leverages user behavior to make recommendations. Collaborative filtering learns what a user likes, reviews, or watches (i.e. depending on whether it's an e-commerce or streaming movie/content site) and based on

interactions with those items, it then recommends products that similar users have also liked, reviewed, or watched. With this approach, collaborative filtering can recommend new kinds of items that a user has not been exposed to before.

Although collaborative filtering has the benefit of providing variety in its recommendations, one of the drawbacks of collaborative filtering is that it suffers from the **cold start problem**. The cold start problem is an issue where a recommendation engine (specifically, a collaborative filtering recommendation engine) cannot recommend a new product that does not have past interaction nor recommend items to a new user without interactions. Content-based filtering, however, does not suffer from the cold start problem. Understanding how new users and products will interact with existing recommendations is essential when building a recommendation engine.

---

*“Not so surprisingly, if you don’t have knowledge of your users, you can’t personalize them. And having no personalization is a huge issue because you want to make new visitors feel welcome so they’ll become loyal returning customers. Repeat customers are ideal and you’ll want to keep them happy, but there’s nothing like adding a new one to the list. This problem is so big that it has a name—it’s called cold start.” (Falk, 2019).*

---

Within the broader ecosystem of recommendation engines, collaborative filtering is further subdivided into two categories called memory-based and model-based methods (Rocca, 2019). These two methods differ in how they determine similarity among users and items.

**Memory-based methods**, also known as “neighborhood-based methods,” leverage past interactions and identify the most popular or highly rated items from similar users. Memory-based collaborative filtering typically follows an item-item or user-user based approach.

These approaches try to find similar items or similar users, respectively, by building a feature matrix of user-item interactions that can then be compared using a similarity measure. There are many options in choosing a similarity measure. Some examples include k-nearest-neighbor (KNN), cosine similarity, or Euclidean distance. Regardless of the similarity measure chosen, the idea is that the closest items or users to the seed item/user are identified as the most similar. In this case, to find similar users or items is to compare the rows or columns of the matrix, shown in Figure 9.

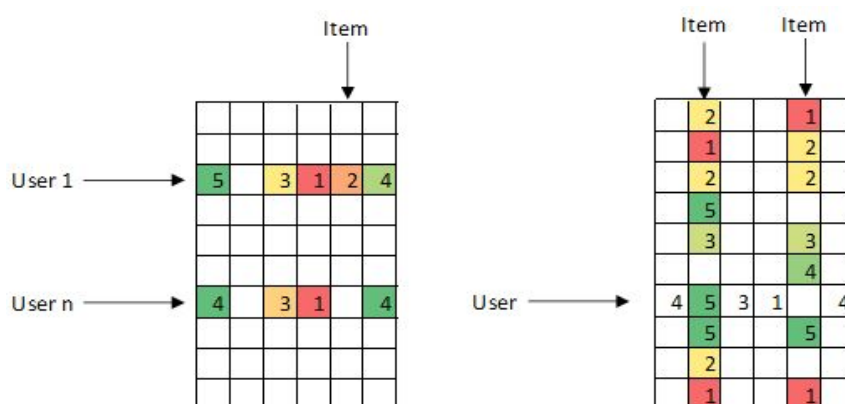


Figure 9: User-user and item-item feature matrices

In both cases, recommendations are made by taking the highest predicted ratings (calculated using a weighted average) for the top  $k$  products that a user has not seen before.

Conversely, **model-based methods** learn from past interactions and attempt to predict the rating or relevance of an unrated item for a user. Model-based methods utilize machine-learning methods to generate a prediction, unlike memory-based methods, which generate predictions using a similarity measure. However, similar to memory-based methods, model-based methods return the items in rank order of highest probability of being 'liked' by a user.

The idea behind model-based methods "is that attitudes or preferences of a user can be determined by a small number of hidden factors," which are considered to be decomposed factors of the user-item interaction matrix (Grover, 2018). This factorization is typically performed via Matrix Factorization (MF) or Singular Vector Decomposition (SVD), but it can also be performed via a deep learning model.

---

*"What matrix factorization eventually gives us is how much a user is aligned with a set of latent features, and how much [an item] fits into this set of latent features. The advantage of it over standard nearest neighborhood is that even though two users haven't rated any [of the same items], it's still possible to find the similarity between them if they share similar underlying tastes" (Luo, 2019).*

---

Both MF and SVD are very similar, with MF decomposing the user-item matrix into two matrices and SVD decomposing it into three. An illustrative example of the matrix factorization approach can be seen in Figure 10.

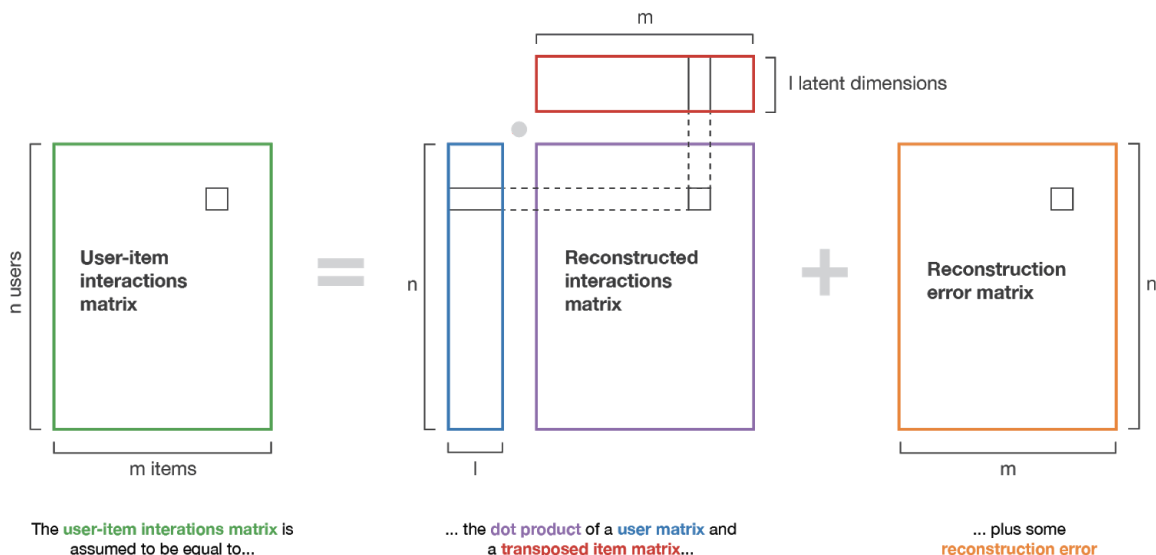


Figure 10: Matrix factorization approach (Rocca, 2019)

Lastly, the **hybrid method** stacks or averages the results of both content-based and collaborative filtering methods, generally in a hierarchical or layered approach. The content-based and

collaborative filtering results can also be used to create an ensemble and predict the item with the most similarity, which will then feed the overall recommendations.

## Modeling Approach

The CognoClick team has spent significant time researching and discussing various modeling approaches to best identify which approach would yield the most relevant recommendations for Amazon's reviewers. This section describes the direction CognoClick took for the POC and outlines the strategy for model development.

Each recommendation engine method has its merits and flaws. **Cold start** is a common challenge for recommendation systems and was considered a key risk that weighed heavily on CognoClick's ultimate modeling approach decision. Per Falk (2019), the best and most robust recommendation engine method against the cold start problem is content-based filtering. Therefore, the CognoClick team knew it had to incorporate a content-based component in order to set Amazon up for success in creating recommendations for new users.

Given the three approaches outlined above of content-based filtering, collaborative filtering, and a hybrid method, **the CognoClick team chose to implement a hybrid method**, as it expects this method will perform better than either a content-based or collaborative filtering method alone. Moreover, a hybrid method will not only mitigate cold start problems but will offer Amazon the scalability it needs for its ever-changing reviewer and product base. Lastly, with a hybrid model, there is additional flexibility to incorporate diverse features, including features around users, products, and review text, a core goal of this POC.

In short, a hybrid method allows CognoClick to utilize the strengths from the various recommendation engine methods and develop a solution that is robust against the cold start problem. More importantly, a hybrid approach best positions CognoClick to deliver high relevance product recommendations for Amazon's reviewers.

While selecting the appropriate recommendation method(s) and implementing them are vitally important tasks, they are only half the battle. One of the main objectives in any recommendation system is to generate **relevant** product recommendations to a user. Although it is possible to train and test the accuracy of product recommendations using the product ratings, the relevance of the actual recommendations themselves (products users have not seen before) can be difficult to determine without end-user feedback.

Therefore, the CognoClick team carefully developed a strategy to help evaluate the relevance of its revamped product recommendations, knowing that this information would be captured once Amazon implements the CognoClick models in production. The team's focus on product relevance and end-user feedback is also appropriate, as Amazon has a core mission of customer centricity and creating a best-in-class customer experience.

Because CognoClick is implementing a hybrid method, it was important to consider all possible options for how best to both integrate and evaluate the recommendations from both content-based and collaborative filtering models. A common industry best practice in evaluating recommendation engines is to **A/B test** or, alternatively, **interleave** model output. Figure 11 outlines the differences between A/B testing and interleaving.

Interleaving conceptually is very similar to A/B testing. In the interleaving process, the top k recommendations are randomly drawn and shown to the end user. The end user interacts with these recommendations blindly. User interaction in turn counts as a positive vote in favor of the model that provided the recommendation. The end user is entirely unaware that this process is occurring. Behind the scenes, this feedback loop can help to reinforce the most effective model, while still generating a number of recommendations that may be pertinent.

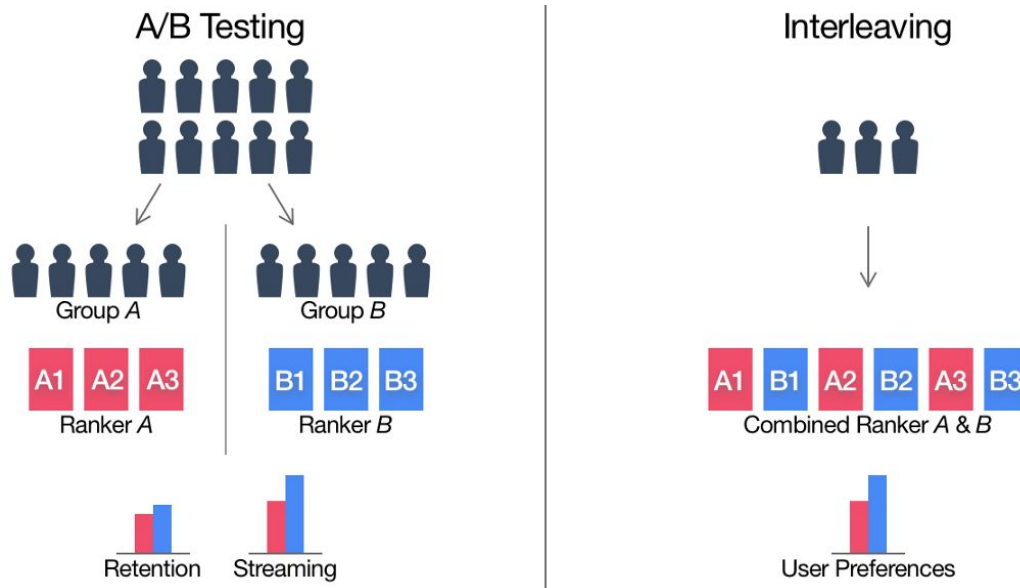


Figure 11: A/B Testing vs. Interleaving (Aurisset, Ramm, Parks, 2017)

Another common industry best practice that the CognoClick team considered is to combine independent models together in an **ensemble**. With an ensemble, the results from both content-based and collaborative filtering models are stacked or averaged together to generate final product recommendations.

Knowing that end-user feedback is critical to the success of a recommendation engine and weighing all the desired goals in the 10-week POC, **CognoClick determined the best approach to deliver on the hybrid method is to interleave content-based and collaborative filtering models.** For all models, the team elected to return the top 10 product recommendation results.

Moreover, because an interleaving approach empowers the agile evaluation of different models, there appears to be almost no downside (besides lack of time) for building as many unique models as possible. Each model can then be effectively implemented in production to provide product recommendations while the user is none-the-wiser to the fact that they are validating and testing multiple models in real-time.

Understanding the opportunity to test and leverage multiple models, the CognoClick team identified three independent models as feasible for the POC.

The three models that were selected for CognoClick's recommendation engine include:



1. **Baseline model**, using product ratings data only.
2. **Review text model**, using features derived from review text.
3. **Deep learning model**, using metadata and text features from both reviewers and products.

The modeling was performed in two phases which are called the **Phase 1 Build** and **Phase 2 Build** for the sake of simplicity. The Phase 1 Build involved creating prototypes for all three models and then scaling each model if and when possible. During the Phase 2 Build, the CognoClick team leveraged the learnings, outputs, and findings from the Phase 1 Build to generate model enhancements. Figure 12 below illustrates these two phases and provides a preview into how the models were leveraged together.

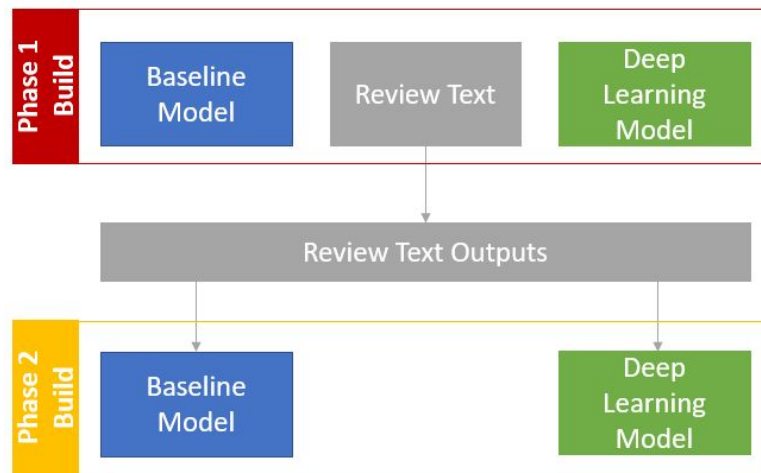


Figure 12: CognoClick Model-Building Strategy

By building three independent models and invoking a phased approach, the CognoClick team enabled agile development of various techniques that were able to be leveraged across workflows.

During the Phase 1 Build, prototypes of each model were built in isolation off of a shared preprocessing procedure. Each prototype model was built using a small sample of reviews/reviewers to ensure that the program would run without abend when scaled. If the program was able to successfully run without abend, then the full data set was passed to the model for training and testing.

Where appropriate, useful model outputs and byproducts were integrated into the preprocessing routine or became inputs into the Phase 2 Build. For instance, the review text model did not require additional development after it was created during the Phase 1 Build. This model's outputs then became inputs into the models for the Phase 2 Build, helping to enhance the Phase 1 models.

In addition to providing robustness, agile development methods also enabled CognoClick to compare and evaluate the logistics of building and running each model against one another. Each model has its own unique technical and technology demands that, in some cases, proved to be very limiting.

Moreover, by building diverse models, CognoClick was able to effectively identify the best models that were both scalable and delivered on project goals. The incremental and iterative process



enabled the team to continuously advance the models. Each model is explained in greater detail in the Discovery and Results section.

Upon completion of the individual models, an **ensemble approach** was taken in an attempt to draw on the unique strengths of each model while providing coverage for the gaps of each model. Although the team recommends an interleaving approach for its hybrid model, an ensemble approach was still considered to help confirm this hypothesis. The ensemble approach was limited to two user clusters from the review text model (Cluster 0 and Cluster 1) due to the limited return results from the baseline model (described in the Baseline Model Phase 2 Build results section).

The ensemble model was a straightforward rules-based ensemble. Figure 13 provides an overview of the CognoClick ensemble. The ratings from all models were averaged across models and sorted. The results with the highest average rating were evaluated as the best recommendations provided by the ensemble model.

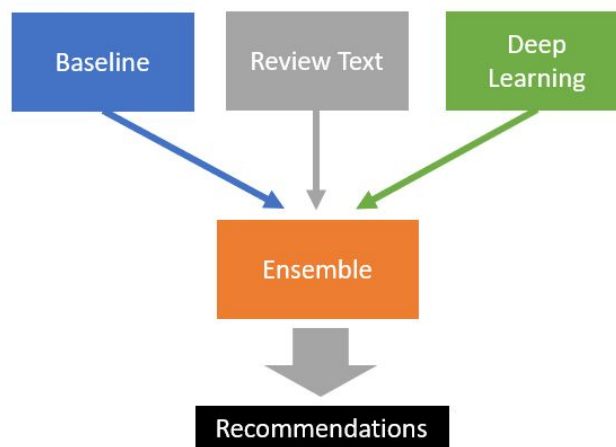


Figure 13: Model Ensemble Overview

Before moving into data discovery and results, there is one specific challenge that should be mentioned: the CognoClick team experienced memory issues and/or long model training time due to the large data set size (in terms of number of records and complexity of features) chosen for the POC. Interestingly enough, each model had a unique experience as it relates to memory limitations that are also discussed in detail below.

To help speed model development across all models and generate results in the Phase 1 Build, the team elected to work with a subset of the electronics data from the Camera & Photo category, which is the second-largest product category within the electronics data.

In filtering the data, the CognoClick team worked with 272,938 reviews from 89,800 reviewers across 12,391 products, reducing the scope by 84 percent to a more manageable load. Figure 14 shows the relationship between the Camera & Photos category and the electronics data.

The recommendation engine models were built leveraging the Camera & Photo data in the Phase 1 Build, and where possible, were reran using the full electronics data set in the Phase 2 Build. When model output is shared, the scope of data is clearly noted.

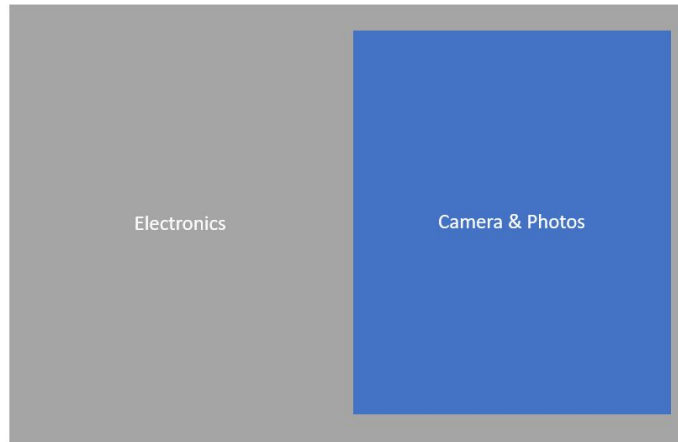


Figure 14: Camera & Photos Relationship to Electronics Data

# Discovery and Results

## Data Preparation

The CognoClick team performed several tasks including data cleaning, data transformation, and feature engineering in order to prepare the data for the recommendation engine. Python was used to complete all analysis and data preparation activities. Although there are three data sets for this POC (product reviews, product metadata, and question and answer data), the data was divided into three different domains to facilitate modeling:

1. **Reviewer features**, which contain information about reviewers.
2. **Product features**, which contain information about products.
3. **Review text features**, which contain topics and sentiment from product reviews.

Each data domain had unique data preparation and transformation activities that are described in greater detail below.

### Reviewer Features

The product reviews data required minimal data cleaning in order to be used for modeling. Across nine total variables, only 'reviewerName' had missing values (approximately one and a half percent of values were missing). Given the nature of this variable and its limited use for modeling, the need to cleanse was unnecessary and resulted in the removal of this variable.

One of the variables, 'helpful', which indicates if a review was helpful or not, was provided in a list format (e.g. [3, 5]). In order to understand this variable in context, the CognoClick team decomposed the list into two separate columns: 'helpful\_numer,' corresponding to the numerator value (the number of up-votes), and 'helpful\_denom,' corresponding to the denominator value (the number of total-votes).

Once the two helpful variables were created, the team then computed a ratio of these values in order to understand the helpfulness of each review. Figure 15 below captures the distribution of the 'helpful\_proportion' across all reviews where a helpful remark was given. For instances where 'helpful' is greater than zero (approximately 37 percent of the data), the vast majority of users find the reviews in the electronics segment to be helpful.

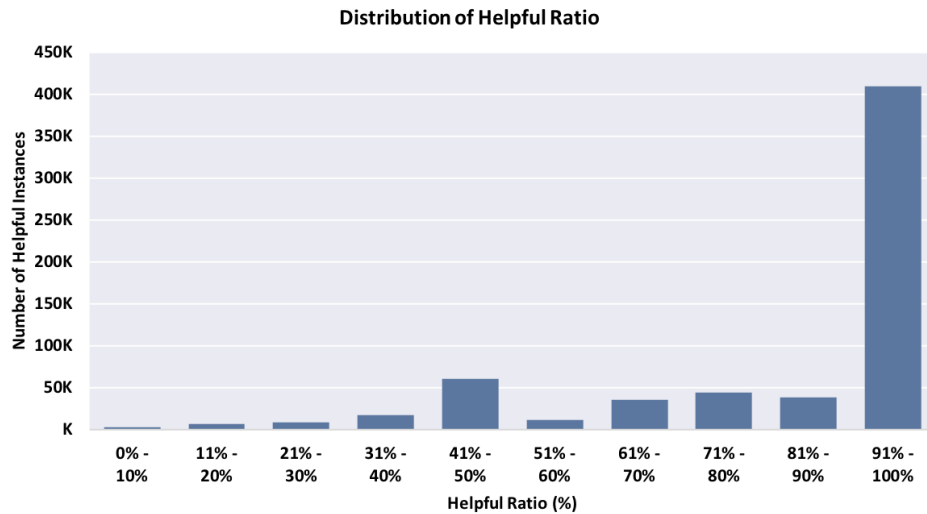


Figure 15: Distribution of Helpful Ratio

In order to understand reviewer behavior, the CognoClick team grouped the reviews data at the reviewer level and computed aggregate statistics for different variables, such as ratings, price, and time between reviews. The ‘duration of time between reviews’ is a derived feature which considers the number of days between each sequential review completed by a given reviewer. The product metadata was also appended to each review in order to capture the price of products purchased at a reviewer level. Figure 16 captures some summary statistics for the reviewer features.

Summary Statistics At Reviewer Level	Mean	Median	Minimum	Maximum
Product Rating	4	5	1	5
Product Price	71	26	0	1,000
Duration Between Reviews (Days)	140	10	0	5,087

Figure 16: Reviewer Summary Statistics

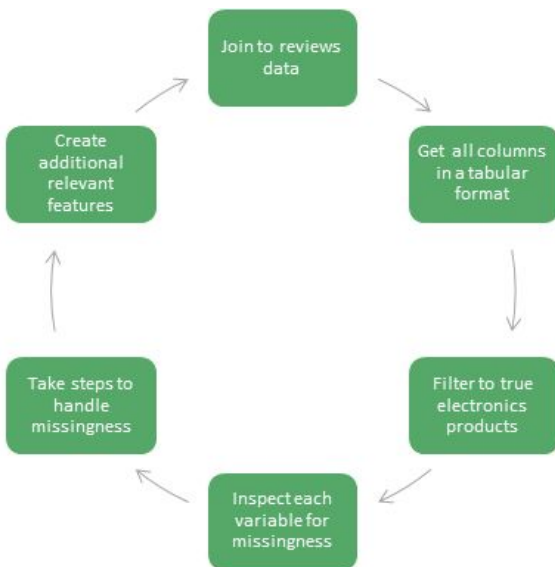
Lastly, Figure 17 shows the distribution of reviewers by their average product rating. At a reviewer level, the average rating exhibits more volatility when compared to the average rating at the product level (see next section for comparison). Knowing that there is more volatility in reviewer ratings is extremely helpful context to have and helped inform modeling decisions related to determining the similarity measure. Table 2 in the Appendix contains a listing of all reviewer features considered for the recommendation engine.



Figure 17: Average Product Rating Distribution for Each Reviewer

## Product Features

The product metadata required extensive data cleaning to be used for modeling. The figure below provides an overview of the data preparation process that this section describes in detail.



First, across the 498 thousand available products, only just over 63 thousand products had a review. One of the first steps in the product data preparation process was **joining the product data to the reviews data** to ensure that only products with reviews were analyzed in greater detail.

Next, some of the key variables, such as 'categories' and 'salesRank', were in a list and dictionary format, respectively. Therefore, these two variables had to be spread wide in order to **achieve a tabular format** that was consistent with the rest of the product variables, which created additional variables for inspection.

---

*"The Amazon Sales Rank is a number which captures the item's popularity in a certain category. The Sales Rank interval can be between 1 and 1 million+. A higher number means you are not getting a lot sales while a smaller number shows that your product is selling well" (Jeane et. al, 2019).*

---

After this initial step, it was clear that the **data needed to be filtered** down to only include products from the top-level categories of 'Electronics' and 'Cell Phone & Accessories' (see the Data Overview section for more information). Using the top-level product category, out of six possible categories, for filtering helped to remove outliers.

Once ‘categories’ and ‘salesRank’ were in a tabular format, **all product variables were inspected for missingness**. Because ‘salesRank’ contains sales rank information for all of Amazon’s product categories, not just electronics, only the electronics sales rank (‘electronicsSalesRank’) was used because this POC is using electronics data. Figure 18 below showcases the product data missingness across the 62,990 products in scope for the POC.

From Figure 18, it is clear that because ‘category6’ and ‘electronicsSalesRank’ contain a high percentage of missing values, these variables were removed from consideration.

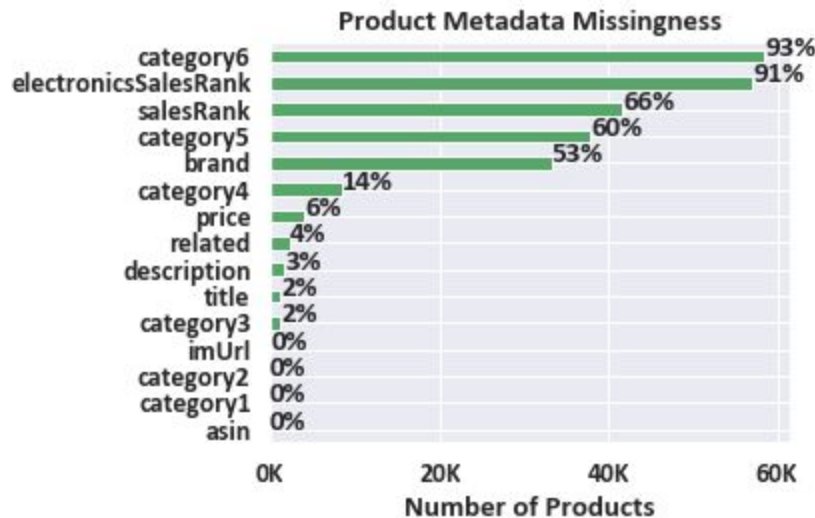


Figure 18: Distribution of Product Variables by Percent of Missingness

As a next step, **each variable was inspected to address missing values** and perform additional data transformation activities.

For each product category variable, if a category value had less than 100 products within it, then these values were combined to form an ‘Other’ category. This step was performed to alleviate sparsity within the data. Furthermore, missing values were addressed by creating an ‘Unknown’ category.

After inspecting each of the product category variables, it was evident that some of the product category variables had too many levels. For example, ‘category3’ had 88 levels, while ‘category4’ and ‘category5’ had just under 200 levels. For simplicity, ‘category2’ was chosen to keep in the final data set, as it only had 14 unique levels. While some of these levels could have been combined together to form less granular levels, the levels were largely left in their raw form since they represent Amazon’s product hierarchy. Figure 19 below shows the second-level category distribution.

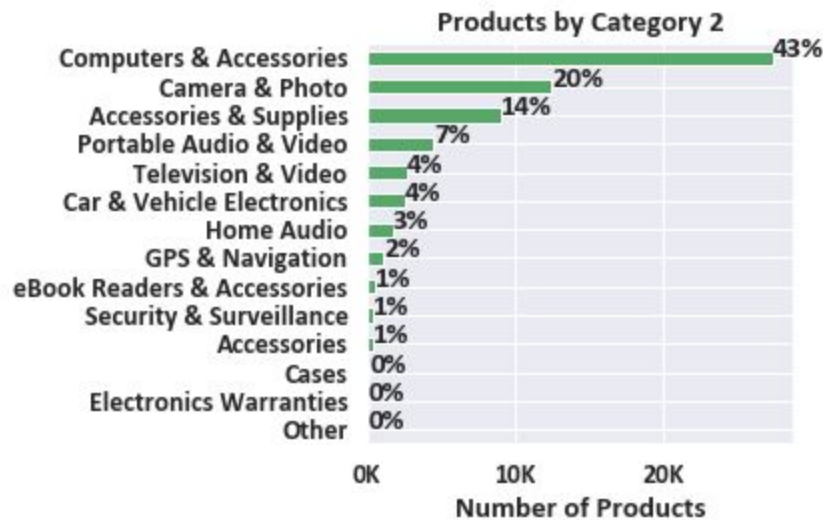


Figure 19: Distribution of products by second-level category

The 'price' variable had some missing values. While creating a binned variable was considered, the price variable was left in its numeric form and missing values were set to 0.

After getting the 'salesRank' variable into a tabular format and discovering the high percentage of missingness within the 'electronicsSalesRank' variable, the original 'salesRank' variable was recoded into a binary variable. In its raw form, the salesRank contained ranks across 29 product categories. The new field, labeled 'anySalesRank,' was kept in scope as it was thought to be a very simplistic proxy for popular products, as a product should have a salesRank if it is being viewed at all.

There were also a few variables that were not helpful. First, the 'brand' variable contained the brand of a product. There are 3,525 unique values within this variable on top of a 53 percent missingness across all products. Therefore, the 'brand' variable was removed from scope.

Second, 'imURL', or product URL, was included in the data set. The URL was a helpful component to the dashboard and mobile application but was not needed for modeling, so it too was removed from scope.

Third, the 'related' variable, which contains details around products that were also viewed or bought, was also dropped from consideration. Although this information is very valuable, it was unclear if relevant features could be derived from this variable to include within a model. Thus, 'relevant' was removed.

Finally, **a few other features were created** to add to the final cleaned products data set. Since product description was included within the data, a binary variable for 'hasDescription' was created.

The products data was also joined to the reviews data to create features for the number of reviews for each product, its average star rating, and distribution by each star (i.e., 1-star, 2-star, etc.). Figure 20 shows the distribution of products by their average product rating. Most products have a high (4- or 5-star) average rating and the distribution is smoother when compared to the distribution of average ratings for reviewers.



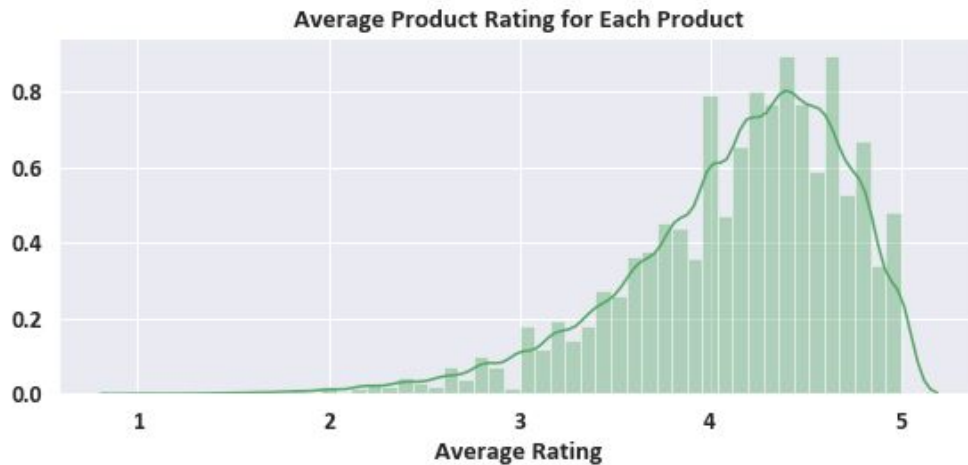


Figure 20: Average Product Rating Distribution for Each Product

The products data was also joined to the question and answer data so that a feature for the number of questions/answers could be created. The final variables for the products data are outlined in Table 3 in the Appendix. All variables contain 62,900 non-null values.

## Review Text Features

**Natural language processing (NLP)** techniques were performed on the reviews data to explore and generate features that have the potential to enhance the overall product recommendations. The reviews dataset as described above contains a rich **corpus** of review text. A corpus is a collection of documents (Lane, Hapke, Howard, 2019).

These review texts are relatively raw containing punctuation, stopwords, and non-alpha characters. During preprocessing, words smaller than 3 letters or larger than 21 letters were removed from the corpus in order to omit many common insignificant filler words, or words that have been accidentally concatenated together. Stopwords were also removed and the words tokenized.

---

*“The different units into which you can break down text (words, characters, or n-grams), are called tokens, and breaking text into such tokens is called tokenization. All text-vectorization processes consist of applying some tokenization scheme and then associating numeric vectors with the generated tokens” (Chollet, 2018, p. 180).*

---

The reviews corpora were generated as a collection of all reviews at both a reviewer-level and a product-level. This approach enabled the CognoClick team to leverage individual reviewer writing styles as features for the model.

In analyzing the reviews at the product-level, it also gave the team the ability to explore item-item similarity and identify product nuances not captured from metadata features alone (i.e., price, number of reviews, etc.). The process for preparing the data at the reviewer-level and

product-level is similar and is described in greater detail below. Figure 21 also provides a summary of the NLP data preparation process.

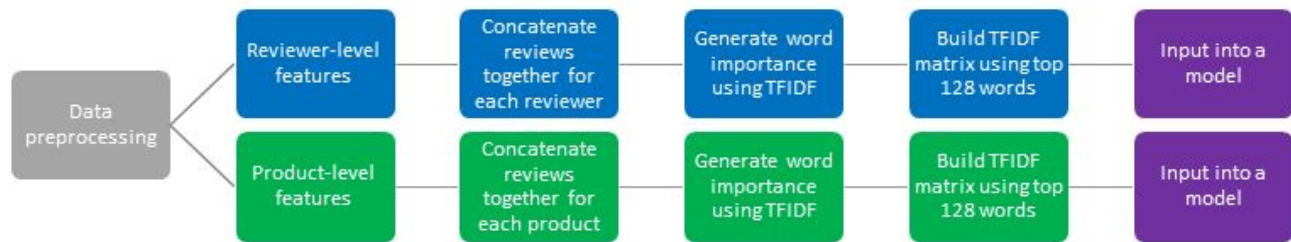


Figure 21: NLP Data Preparation Overview

To create reviewer-level features, the reviews were first concatenated together by reviewer. Collapsing the data this way resulted in 192 thousand rows, which represent the number of unique reviewers within the electronics product category. This process created one record or row per reviewer and two columns: the reviewer identifier and all of the individual's reviews strung together.

The same exercise was performed for reviews by product. The resulting data frame contained one record per product and two columns: the product identifier and all of the reviews associated with that product strung together.

Once the corpora were successfully prepared, each resulting dataset was passed to the Sklearn implementation of **Term Frequency, Inverse Document Frequency (TFIDF)**. TFIDF is a widely used method for evaluating and generating the value of each word in a corpus of documents. TFIDF is effective because it generates a representation of both the frequency and the uniqueness of words in a corpus (Lane et al., 2019). The resulting TFIDF matrices are able to serve as inputs for modeling, both for clustering and for as potential embeddings for deep learning models.

To enable the model to be fit readily into memory on most machines, the **vocabulary**, or set of valid tokens, was limited to the top 128 words (Lane et al., 2019).

The max **n-gram**, or number of separate tokens considered together (e.g. ice, cream separately versus ice cream), length was initially set to one ( $n=1$ ) for simplicity's sake in the initial analysis (Lane et al., 2019).

**Bi-grams** (when  $n=2$ ) appear more likely to return stronger results by ensuring that “not recommend” and “low quality” do not mistakenly get tokenized as “recommend” and “quality.” It was the bi-gram implementation that was ultimately passed to the TFIDF matrix and served as input for other models.

## Results

With the three data domains (reviewer features, product features, and review text features) prepared, the team was ready to start building its recommendation engine for Amazon.

The three models in scope for the POC are described in great detail in this section. For each model, the necessary background, results, and the process to generate actual product recommendations for reviewers are discussed. Because CognoClick chose to proceed with a phased modeling approach, each model features a section for the Phase 1 Build and Phase 2 Build.

As a refresher, the three models in scope are:

1. **Baseline model**, using product ratings data only.
2. **Review text model**, using features derived from review text.
3. **Deep learning model**, using metadata and text features from both reviewers and products.

## Baseline Model

### Background

In determining how best to build a baseline model, only collaborative filtering methods were considered, since these methods offer greater variety in recommendations to a user. Both memory-based and model-based methods for collaborative filtering were built to help determine a 'best' baseline model.

With the broader goal of wanting to leverage both metadata and text features in the final model, it was clear that the purpose of the baseline model was to serve as a point of comparison. The baseline model would help show the value in adding more complex features and in leveraging a more complex deep learning model.

Therefore, rather than build the baseline models from scratch, the Python Surprise package was leveraged for baseline model building. The Surprise package has the advantage of offering many recommendation algorithms along with features like cross validation and parameter tuning in a very accessible format, which enabled the team to devote more time to NLP processing and building a deep learning model. Moreover, the Surprise package allowed for a model-based approach to be considered as a baseline model.

---

***“Surprise is a Python scikit building and analyzing recommender systems that deal with explicit rating data. The name SurPRISE (roughly :) ) stands for Simple Python Recommendation System Engine” (Hug, n.d.).***

---

Two different algorithms were used to build a baseline model: k-nearest neighbors (KNN) and Singular Value Decomposition (SVD). **KNN** is a simple machine-learning algorithm that uses a distance measure (memory-based) to determine similar groups of users/items and return the top k nearest neighbors from a given user/item. For this problem, an item-item collaborative filtering method was used, since ratings for items are thought to be more stable over time compared to the tastes of users (Luo, 2019). Thus, KNN computes a distance between a given product and all other products, ranks these distances, and returns the top k products that are most similar.

**SVD** was described in detail in the Recommendation Engine Methods section and decomposes the user-item matrix in an attempt to predict ratings for products that a user has not rated (model-based).

For both types of collaborative filtering models, hyperparameter tuning was performed using three-fold cross validation. This approach ensured that an optimal model was built for each method. With the KNN model specifically, adjustments were made to incorporate the average rating for each user, which helped to normalize the ratings data and account for the different preferences of each user (as observed from the Data Preparation section). Through hyperparameter tuning, Pearson correlation proved to be the optimal distance measure.

Several different metrics were used to evaluate the models. **Root mean squared error (RMSE)** and **mean absolute error (MAE)** are common metrics to evaluate recommendation engine performance, as they help to show in different ways on average how far the predicted ratings are from actual ratings and have the added benefit of using the same scale as the ratings (lower value is better). **Fraction of Concordant Pairs (FCP)** is a metric that analyzes the number of correctly predicted item rankings over all predictions, similar to an R-squared metric (higher is better).

Furthermore, **precision and recall at k** are also common metrics to use in evaluating recommendation algorithms and were used to evaluate the baseline models. Since precision and recall are metrics that evaluate the accuracy of a classification model (binary output), the predictions were first transformed to binary output. This transformation can be defined as follows: An item is relevant if it has a true/actual rating  $\geq 4.5$  and a recommended item is considered relevant if it has a predicted rating  $\geq 4.5$ . The threshold of 4.5 was chosen for simplicity given most of the ratings are positive (the average rating is 4.22 and median rating is 5.0).

The precision and recall at k metrics evaluate recommendation accuracy in a different way. Precision at k examines the proportion of relevant predictions (predicted rating  $\geq 4.5$ ) within all predictions when a different number of k items are returned, answering the question, “how relevant are my recommendations?”

Recall at k, on the other hand, examines the proportion of relevant items over the total number of relevant items at different levels of k, answering the question, “do my recommendations contain high relevance?” Precision and recall represent a tradeoff, so models with both high precision and high recall at k are generating extremely relevant recommendations.

Lastly, the baseline models were built using data from the Camera & Photos category to help speed model development and generate results. The only inputs to the models were the reviewer id, product id, and product ratings. An 80/20 split was used to create the training and test sets for each model.

## Phase 1 Build

During the Phase 1 Build, the baseline models were built using ratings reviewers had given to Camera & Photos products. Figure 22 shows the test set results from the two baseline models, KNN and SVD. Three-fold cross validation results can be found in the Appendix - Table 4.

	RMSE	MAE	FCP	Avg Precision at k	Avg Recall at k
KNN	1.093	.809	.577	.886	.556
SVD	1.022	.766	.546	.922	.549

Figure 22: Baseline Models Test Set Results (Camera & Photos Data)

Given the above results, the SVD model has a slight edge, although both models have sub-par performance. With high MAE values and RMSE values around one, the current predictions are almost a whole star-rating off, which is significant. In addition, the FCP values are not close to one, and interestingly the KNN model does a slightly better job of preserving item ranking.

Furthermore, both models have high average precision at k (three-fold cross validation was used to compute these results), yet low average recall at k. This result means that the models are not capturing true relevance since they are not recommending items that are actually/truly relevant.

Overall, it was clear from the Phase 1 baseline model results that there was room for improvement in recommending relevant products to reviewers. From the Phase 1 Build, the SVD model was determined to be the better model and is the model that was used in the Phase 2 Build.

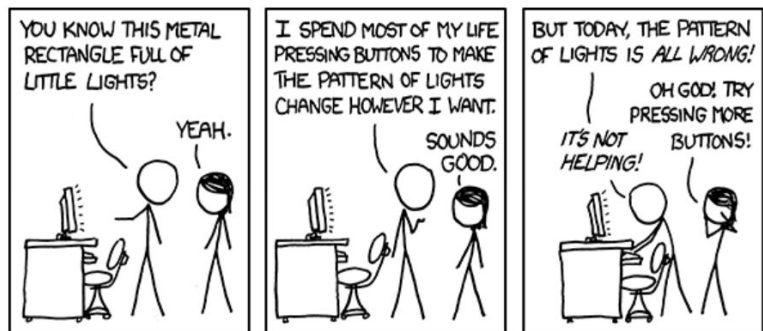
## Phase 2 Build

The main objective in the Phase 2 Build was for the baseline model (SVD model) to generate recommendations for products users had not rated before. In order to meet this objective, additional data preparation steps were required.

First, the model was fit to the full Camera & Photos data set (products users had rated) and saved for later use. Next, the data was reformatted so that all the products that users had not rated were in a single data frame. While most modeling techniques require a matrix format for the data, with products on rows and users on columns for instance, the Surprise package requires a certain data frame format for modeling. Therefore, work was done to conform to this format.

Unfortunately, memory issues were encountered in trying to reformat the data on a single desktop machine. Therefore, Google Cloud Platform (GCP) was leveraged to help alleviate the memory issues. The GCP server had 24 processors and 650 GBs of RAM. Although progress was made to get past data formatting issues, there were additional memory issues in trying to generate recommendations.

The Surprise package also does not support parallelization, so extra time was spent to try and scale GCP resources to successfully generate the product recommendations.



GCP was scaled beyond what many enterprises have available for data science problems, but the available GCP resources still could not handle generating recommendations across every user and every unseen product within the Camera & Photos dataset. **This approach for the baseline model had to be abandoned because of the memory issues encountered.**

These memory hurdles are typical for recommendation engines and highlight the importance of having an environment with ample resources to appropriately scale the data preparation and modeling work.

After taking a step back, the CognoClick team more clearly recognized that the baseline model was memory-intensive and had high technology demands. The team made the determination that the baseline model had to be run on smaller subsets of data or it would never be commercially viable.

Thus, the team moved forward **using the baseline model in conjunction with the user clusters from the review text model** (see Review Text Model section below for more information). Running the model on the user clusters presented a smaller scope of users and unseen products for the model, and the hope was that the recommendations would actually finish. Thankfully, recommendations were able to be generated on two clusters, and the top 10 product recommendations for each reviewer was returned.

The top 10 product recommendations from the initial clusters were inspected in an attempt to validate that the baseline model was returning relevant recommendations. For both clusters, all product recommendations had a predicted rating of 5 (out of 5), which was not expected. Although the results were different for each reviewer within a cluster, it was unclear if the model was actually discriminatory in its results.

Once again, the team took a step back to understand these results. For one, the baseline model was generated using data from the Camera & Photos category only, and the products that were reviewed had very high positive ratings overall. Figure 23 shows the distribution of ratings within the Camera & Photos category as well as a high level summary of the user clusters (based on actual activity). From this information, it is clear that the users within these clusters are predisposed to have high predicted ratings.

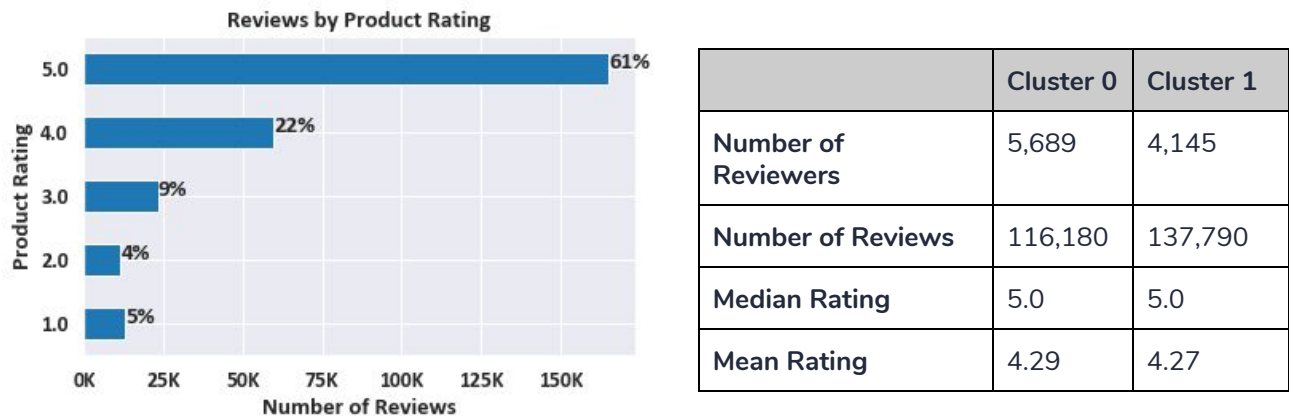


Figure 23: Camera & Photos Ratings Distribution (Left) | Cluster Overview (Right)

Furthermore, within the Camera & Photos dataset, 40 percent of the reviewers only gave a single review. Therefore, less variability within the product recommendations is expected.

Given the lack of confidence in the baseline model to generate meaningful predictions for two of the review text model user clusters, **the CognoClick team decided to not continue forward with implementing the baseline model.**



Overall, while this exercise did not yield usable results, it did help to demonstrate that two separate models can be combined together to solve a business problem. The CognoClick team feels that if more time had been allowed, this model integration would have been successful. In the future, Amazon may need to consider generating specialized models on subsets of users and/or products if results are not as expected. The approach here provides a starting point for that endeavor.

## Review Text Model

### Phase 1 Build

In the Phase 1 Build for the review text model, two separate cluster analyses were performed on reviewers and products to support user-user and item-item similarity analyses, respectively. The large scale of data required CognoClick to scale up to leverage cloud resources (Google Cloud Platform) to successfully complete these analyses.

For each implementation, the text features that were extracted from the review corpus (in the form of a TFIDF matrix) were passed to a **k-means clustering algorithm**. The reviewer or product was assigned to one of ten possible clusters.

Graphically, the clusters formed natural breaks in the two-dimensional representation and the top terms passed the “eye test” for each cluster. The clustering representation leveraged cosine similarity and Euclidean distance metrics to graph these exhibits and displayed cluster variance using **multidimensional scaling (MDS)** and **t-Distributed stochastic neighbor embedding (t-SNE)**. Both MDS and t-SNE are types of dimensionality-reduction techniques to help visualize clusters.

It is very computationally expensive to graph clustering analyses. The product (item) clusters from the Camera & Photos data were the only clusters that were able to be successfully rendered. The reviewer clusters from both the full electronics data and the Camera & Photos data were too complex to be graphically represented.

Despite not being able to be graphically rendered, the cluster analysis at the reviewer level was able to be successfully performed using the full electronics data as well as the Camera & Photos data. The descriptive analysis of the Camera & Photos data is included below. Figures 1 and 2 in the Appendix contain details around reviewer clusters from the full electronics data.

The **reviewer clusters** for the Camera & Photos data is outlined in Figure 24 below. The clusters represent groups of reviewers that have written similar reviews in the Camera & Photos category. The most popular cluster for reviewers is Cluster 7, which contains nearly 25 percent of the total reviewers. The next two clusters with the most members are Cluster 2 and Cluster 9, which contain 17 percent and 13 percent of the total reviewers, respectively.

Label	Cluster 0:	Cluster 1:	Cluster 2:	Cluster 3:	Cluster 4:	Cluster 5:	Cluster 6:	Cluster 7:	Cluster 8:	Cluster 9:
Members	5,688	4,145	15,714	5,918	3,547	10,394	5,891	22,022	4,253	12,228

Figure 24: Distribution of Reviewers in Clusters (Camera & Photos)

Figure 25 displays the top 10 terms that each cluster of reviewers is using in their reviews. Cluster 7 contains high-quality/positive terms that appear to have a positive sentiment. In contrast, all other clusters appear to have a specific product as their lead term. The terms “batteries,” “bag,”



“camera,” “tripod,” “lens,” “video,” and “case” are all unique types of individual camera products and accessories that roll up to the Camera & Photos category.

Figure 25 also suggests that it may be advantageous to reduce the number of clusters from ten to eight, allowing the model to combine Cluster 9 and Cluster 2, as well as combine Cluster 0 and Cluster 3.

Clusters 2 and 9 seem to have key terms that contain positive connotations towards “camera” (the top term in each). However, Clusters 2 and 9 have slightly different supporting terms outside of the positive terms. In Cluster 2, “pictures” and “picture” appear compared to “light” and “flash” in Cluster 9. It may stand to reason that the reviewers in Cluster 9 value the ability of the camera to capture lighting, while Cluster 2 is more focused on the pictures themselves.

Clusters 0 and 3 are focused on camera batteries and camera quality. It is difficult to draw any conclusions on the differences between Cluster 0 and Cluster 3 without additional descriptive analysis of the underlying data in each cluster. However, it is possible that Cluster 0 represents purchasers of after-market battery accessories, while Cluster 3 represents reviewers who care about the battery life of the camera they have reviewed, or visa-versa. Further analysis could help determine if the clustering procedure needs to be modified to ensure as distinct clusters as possible.

Label	Cluster 0:	Cluster 1:	Cluster 2:	Cluster 3:	Cluster 4:	Cluster 5:	Cluster 6:	Cluster 7:	Cluster 8:	Cluster 9:
1	batteries	bag	camera	battery	tripod	lens	video	great	case	camera
2	charger	camera	pictures	camera	camera	camera	camera	good	camera	flash
3	battery	lens	great	charger	use	great	quality	product	fit	one
4	camera	great	use	one	great	canon	good	works	great	lens
5	great	well	good	works	one	use	use	use	well	use
6	use	one	one	canon	well	lenses	great	one	good	get
7	one	fit	cameras	great	would	good	one	well	strap	like
8	good	strap	take	good	good	one	get	price	one	light
9	well	like	quality	use	like	nikon	like	would	use	canon
10	work	would	picture	batteries	price	get	easy	quality	would	good

Figure 25: Key Terms Identified by Reviewer Cluster (Camera & Photos)

Next, the **product clusters** were examined in greater detail. Figure 26 contains the distribution of products across the Camera & Photos category. The most popular cluster is Cluster 2, which contains 23 percent of all products. Cluster 7 contains the fewest products and represents just over two percent of all Camera & Photos products.

Label	Cluster 0:	Cluster 1:	Cluster 2:	Cluster 3:	Cluster 4:	Cluster 5:	Cluster 6:	Cluster 7:	Cluster 8:	Cluster 9:
Members	504	662	2,841	1,567	679	1,724	653	320	621	2,820

Figure 26: Distribution of Products in Clusters (Camera & Photos)

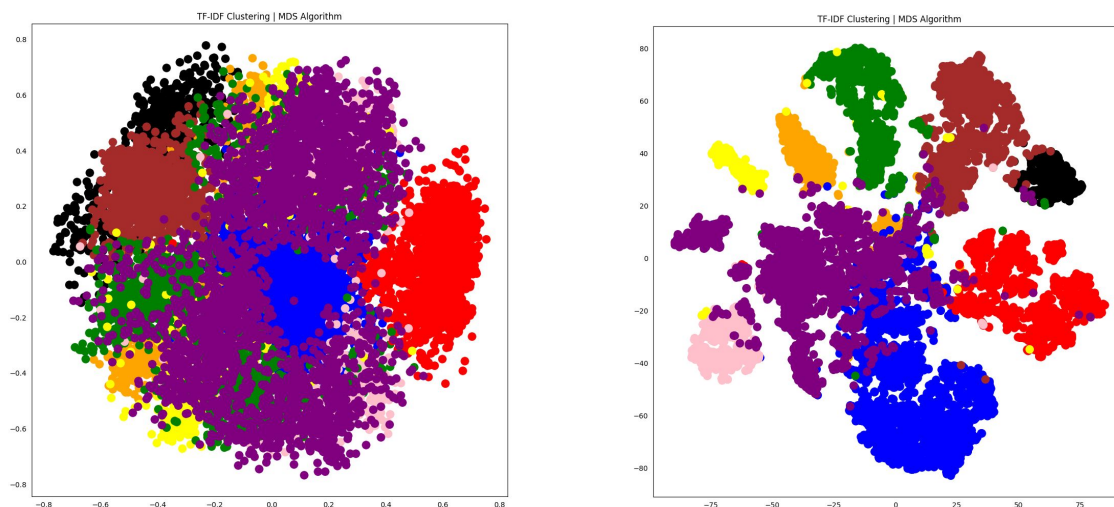
Similar to the reviewer clusters, the top 10 terms across the reviews of each product cluster is shown below in Figure 27.

Label	Cluster 0:	Cluster 1:	Cluster 2:	Cluster 3:	Cluster 4:	Cluster 5:	Cluster 6:	Cluster 7:	Cluster 8:	Cluster 9:
1	filter	tripod	camera	lens	bag	battery	flash	strap	case	use
2	lens	camera	video	lenses	camera	batteries	light	camera	camera	one
3	quality	use	pictures	canon	lens	charger	use	one	strap	light
4	use	one	cameras	focus	strap	camera	camera	use	fit	great
5	good	well	use	camera	fit	one	one	like	bag	screen
6	lenses	would	good	use	lenses	canon	works	would	well	good
7	one	great	quality	nikon	case	works	great	around	lens	camera
8	great	like	one	good	small	great	well	well	would	would
9	well	good	great	one	one	price	canon	tripod	small	well
10	camera	get	zoom	great	well	good	like	great	extra	get

Figure 27: Key Terms Identified by Product Cluster (Camera & Photos)

The key terms for product clusters in Figure 27 are consistent with what one might expect. The top term for each cluster is a unique camera item or accessory, such as “tripod” or “lens.” The rest of the key terms for each cluster generally supports the top term in each cluster. For example, in Cluster 0, the top term of “filter” is followed by “lens” and “lenses.”

To continue the analysis of the product clusters, the two-dimensional graphical representation of the product clustering scheme is shown below in Figures 28 and 29.



Figures 28 and 29: Clusters Graphed with MDS (Left) | t-SNE (Right)

Using the review text aggregated by product to create clusters successfully split the products into distinctive groupings. The t-SNE (right) graphic contains the better representation. Each cluster contains similar products that might be recommended to a user based on their past purchasing history or recent activity.

Cluster 9 (purple) has several words indicating quality and favorability in the grouping, but without a distinctive product-like term (other than camera). This result may be contributing to the less distinct nature of this cluster, causing it to overlap with other clusters in some places. This cluster may also be a cluster of high quality general cameras and accessories that should be further analyzed.

Overall, the clustering approach using review text data was successful in leveraging memory-based collaborative filtering methods. Using the unique clusters shown above, it is possible to identify similar users and similar items en route to providing recommendations.

## Phase 2 Build

Ultimately, the intention of performing the TFIDF matrix calculations was to pass the TFIDF matrices for each perspective (products vs. reviewers) to other models as representations of the reviewers and products. Therefore, no additional work outside of the Phase 1 Build was needed for the Review Text Models. The focus in the Phase 2 Build was feeding and/or integrating the review text features and/or clusters into other models.

The user clusters were able to serve as inputs into the baseline model (described above), while both the TFIDF matrices for products and reviewers were fed into various deep learning models. The results of the deep learning models are described in detail below.

Beyond integrating with other models, the review text output contains other benefits for Amazon. The product clusters can be leveraged to produce a “similar items” functionality and create a content-based filtering model. The reviewer clusters can be leveraged to identify similar users based on their reviews. Other users’ highly rated items can then be shared amongst similar users.

An opportunity for further analysis even includes cross referencing the **implicit** review information against the **explicit** numerical ratings to detect relationships between “what is said” and “what is rated.” The CognoClick team may be able to use this new information to identify products that are “hidden gems.” Regardless, it is clear that the review text data presents a wealth of opportunities, which is why it is a foundational component of the CognoClick hybrid approach.

It is worth mentioning that the number of clusters generated by the models were not revised from ten to eight groups. There was enough variation in the clusters upon further analysis to help the team feel comfortable that the distinct clusters were indeed segmented enough to justify not changing them. Additionally, smaller clusters enabled better iteration for the baseline model.

## Deep Learning Model

### Background

Two primary goals of this project were to assess different models and leverage both text and metadata features in a model. A deep learning approach was considered as one avenue to meet these goals due to the large amount of features (for reviewers and products) and product review text available.

In order to reach the desired end state of a hybrid model, the CognoClick team built a collaborative and content-based deep learning model using Keras as described in the Recommendation Engine Method section.

Each model was built iteratively starting with a “bare bones model,” which consisted of only the user identifier, product identifier, and product rating (Phase 1 Build), and increased to ultimately include metadata and TFIDF matrices from the review text (Phase 2 Build). In addition to the varying input features, multiple model structures were assessed to determine which structure yielded the lowest loss and highest accuracy.

In all instances, the deep learning models were built using the full electronics data set. Despite being the most complex in nature, the deep learning models ultimately had the best performance from a technology perspective and were able to be run on a local desktop computer.

## Phase 1 Build

The goal of the Phase 1 Build for the deep learning model was to establish a foundation using ratings data and plan for both collaborative and content-based methods in the Phase 2 Build.

First, a collaborative filtering model was developed. The deep neural network architecture for the baseline collaborative filtering model can be seen in Figure 30 below.

In this model, parallel inputs for reviewer and product features were fed into user and product embedding layers (respectively), which were then trained to predict the rating given by the reviewer for that product.

An **embedding** is a relatively low-dimensional space which can be used to translate high-dimensional vectors (Embeddings, n.d.). Embeddings make it easier to perform machine learning tasks on large inputs like sparse vectors representing words, which the CognoClick team had in the form the product reviews. An embedding can also be learned and reused across models, which CognoClick takes full advantage of in the Phase 1 Build and beyond.

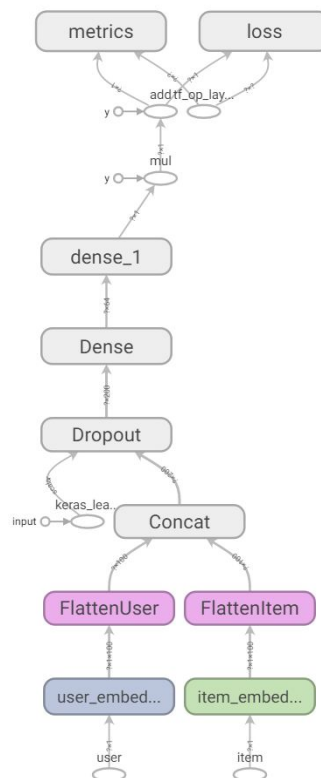


Figure 30: Baseline Collaborative Filtering DNN Model Architecture

Various layers, including multiple dense layers, varying dropout layers, varying learning rates, and a scaled sigmoid output, were assessed and trained to identify the champion collaborative model.

The baseline collaborative filtering architecture helped to provide a framework for this additional model development.

A key component of the collaborative filtering model architecture is the **sigmoid** activation function, also known as a logistic function, which has a smooth curve and returns a value between 0 and 1. This output is trained to predict the rating of the user product pair which is then scaled between 1 and 5 to match the range of the product ratings. A visual depiction of the sigmoid function is shown in Figure 31. The sigmoid function does a good job of predicting the range of values and does so very quickly as compared to other functions or classification algorithms (e.g., multi-class softmax).

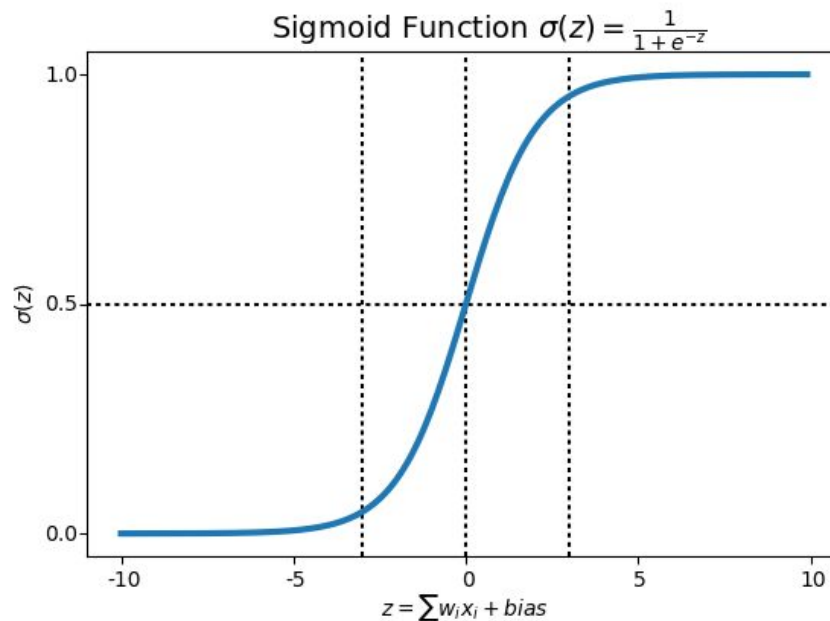


Figure 31: Sigmoid Function

The Phase 1 Build collaborative filtering model performed reasonably well using only the user and product identifiers as input while predicting the user rating as the output. This model had a mean absolute error (MAE) of 0.5929 on the training set (90 percent split) and 0.7435 on the validation set (ten percent split).

This result indicated that the model was predicting roughly one-star off the correct rating as compared to the true rating. While this result could be improved, it offers an improvement over the baseline model (SVD test set MAE was .766). The tensorboard loss and accuracy can be seen in Figures 32 and 33. These results were very promising to the CognoClick team.

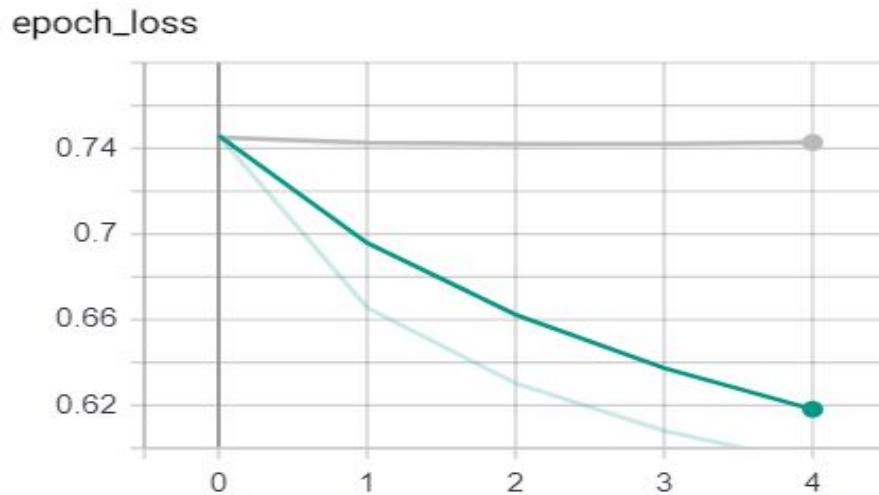


Figure 32: Epoch Loss (MAE)

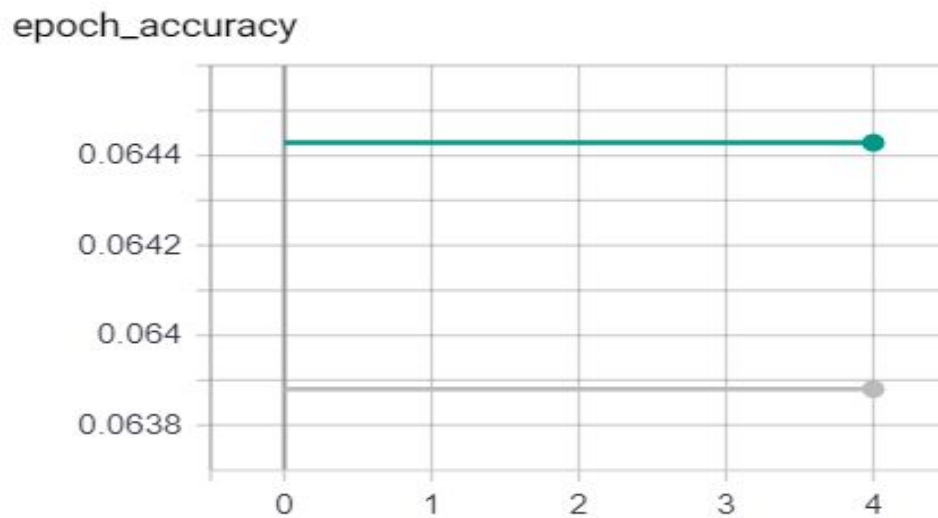


Figure 33: Epoch Accuracy (MSE)

Since the baseline collaborative filtering model created both reviewer (user) and product (item) embeddings, it offered the team the ability to also create a content-based model from these embeddings. An embedding tries to capture some of the semantics of the input by placing semantically similar inputs close together in the embedding space. **For recommending similar products, an embedding is an effective way to identify products which are close together in the latent feature space.**

The team decided to leverage the product embeddings to perform a similarity analysis and identify similar products to a seed product. However, this process proved to be very challenging. First, the size of the embeddings was too large for in-memory calculations using the Python sci-kit learn package. This result meant that additional research for enhanced performance measures was required and ultimately performed using Scipy.



After running the similarity analysis using cosine as the distance measure, it was unclear if the embeddings were providing relevant results in the form of similar products. Additional research was required to assess the optimal approach for the Phase 2 Build.

## Phase 2 Build

The Phase 2 Build included a robust and exhaustive assessment of deep learning model architectures and performance, building upon the Phase 1 Build results.

Initial efforts were focused on incorporating additional metadata into the models, since the Phase 1 Build only leveraged the product ratings and user and product identifiers. The metadata features considered were at both a reviewer- and product-level and included information around the price, number of reviews, and ratings. In addition to metadata features, review text features in the form of TFIDF matrices were also incorporated into the models. The Data Preparation section contains full background on the features available for modeling.

After extensive model training, the champion collaborative filtering model included dropout layers on five embedding and four dense layers. The deep neural network architecture can be seen in Figure 34 below. The model has eight inputs which include: user identifier, item identifier, category identifier, review year, review month, user and product metadata, and TFIDF vectors. These data points are fed into the model separately, then concatenated as the data moves through the various fully connected (dense) layers.

This champion collaborative filtering deep learning model was used for three purposes:

1. Identify similar reviewers using user embeddings
2. Identify similar products using product embeddings
3. Predict ratings for user and product combinations

Each of these three approaches can be used independently or combined as part of the hybrid model. The team ultimately leveraged the champion collaborative filtering model for purposes two and three.

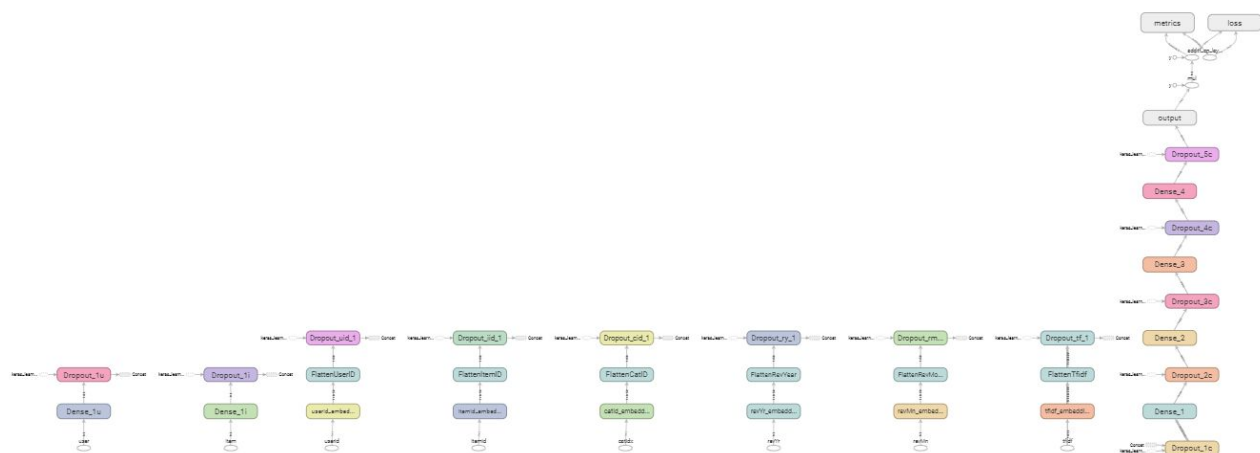


Figure 34: Champion Collaborative Filtering DNN Model Architecture

The champion collaborative filtering model improved the baseline loss and validation scores to a MAE of 0.5800 and 0.6620, respectively. The training loss improved by 0.02 while the validation loss improved by nearly 0.13, which equates to a 16 percent improvement. The champion



collaborative filtering model loss and accuracy results can be seen in Figure 35 below. Adding in additional features helped to increase the accuracy in the product recommendations that were made.

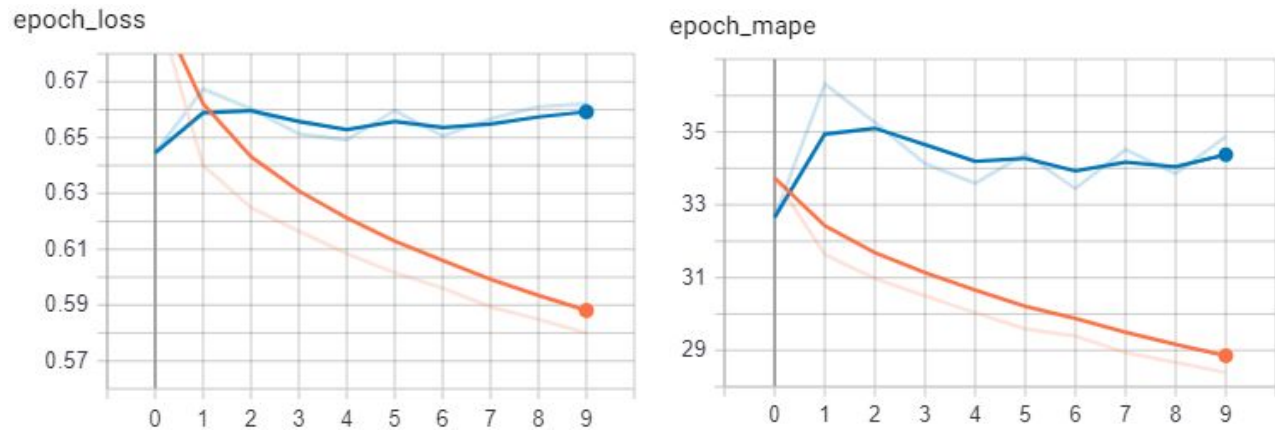


Figure 35: Tensorboard Loss and Accuracy Results

With the collaborative filtering model built, the team needed to regroup on the content-based model in order to deliver on its desired hybrid approach. During the Phase 1 Build, the CognoClick team found that comparing product similarity using cosine distance was inconclusive about its effectiveness. Additional research and testing was performed which determined that leveraging Euclidean distance provided to be a better measure of similarity based on subjective assessment of the results.

While the similarity results from the content-based model were improved using Euclidean distance, it was still unclear if this model was capturing the latent semantics of the products or user preferences. Since a collaborative model learns the user semantics that are unique to a user's preferences, those preferences could be accounting for the product variability. This personal preference (into an individual user's tastes) proved to be very difficult to understand and impacted the team's ability to assess the relevance of results. Ultimately, the champion collaborative filtering model was leveraged to predict ratings for products not previously reviewed by a user.

To gain confidence in the similarity results, predicted ratings for multiple users were inspected in great detail. It was observed that overall users' predicted distributions varied, which indicated the model was capturing a good representation of user preferences. Histograms denoting two sample user predictions can be seen in Figure 36.

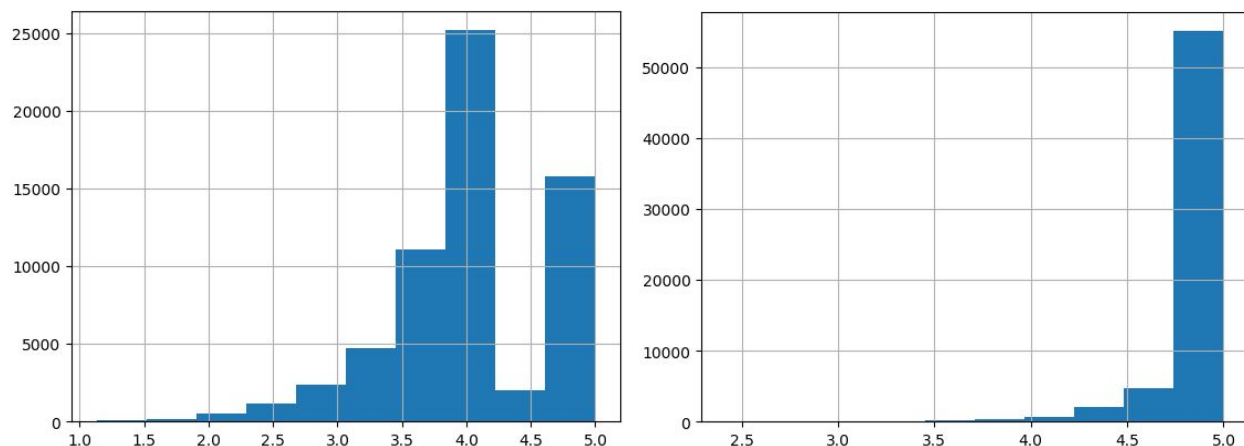


Figure 36: Histogram of Two User Rating Predictions From Euclidean Similarity Model

To offer comparison against the similarity results (memory-based), a second content-based model was developed using an autoencoder. An **autoencoder** is a type of artificial neural network used to learn efficient data codings in an unsupervised manner (Chollet, 2018, p. 296). The goal of an autoencoder is to learn a representation (encoding) for a set of data, typically for dimensionality reduction, by training the network to ignore signal “noise.”

Along with the reduction side (encoder), a reconstructing side (decoder) is learned, where the autoencoder tries to regenerate a representation as close as possible to its original input. A generic example of an autoencoder is shown in Figure 37 for reference.

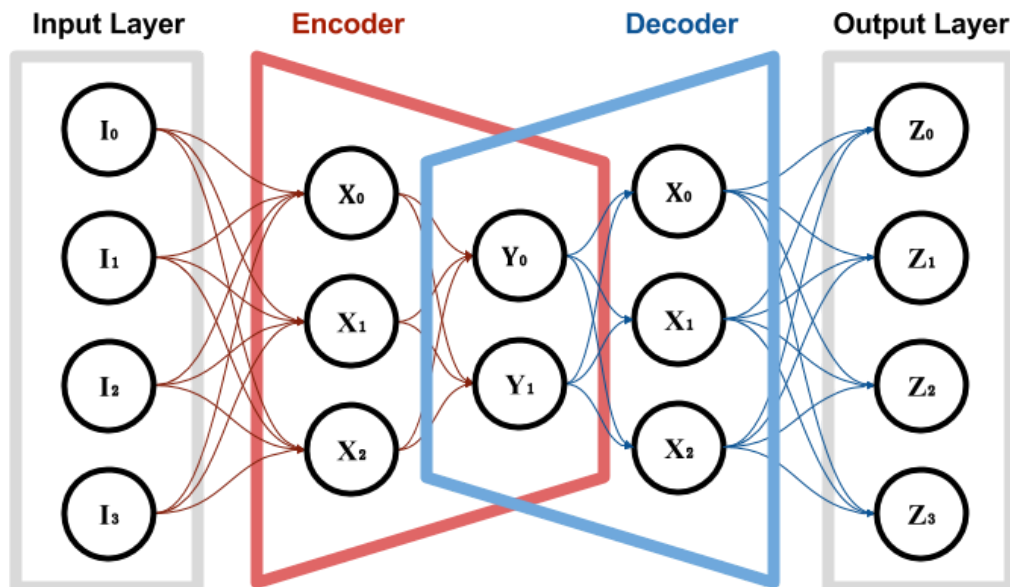


Figure 37: Example Autoencoder Model Structure (Zucconi, 2019)

Similar to the collaborative filtering modeling approach, multiple model iterations using different types of layers (dense and embedding), number of layers, node sizes, and dropouts were assessed to identify the optimal autoencoder architecture.

The architecture of the champion autoencoder can be seen in Figure 38. This model features two dense layers on both sides of the encoded dense layer for a total of five hidden layers without an

embedding layer. The autoencoder performed very well with a MAE loss of 0.0508 and a MAE validation loss of 0.0667, significantly better than any other model. **The extremely low MAE loss results demonstrate that the autoencoder was able to closely encode the data into a dense representation of fifty total features with a high accuracy.**

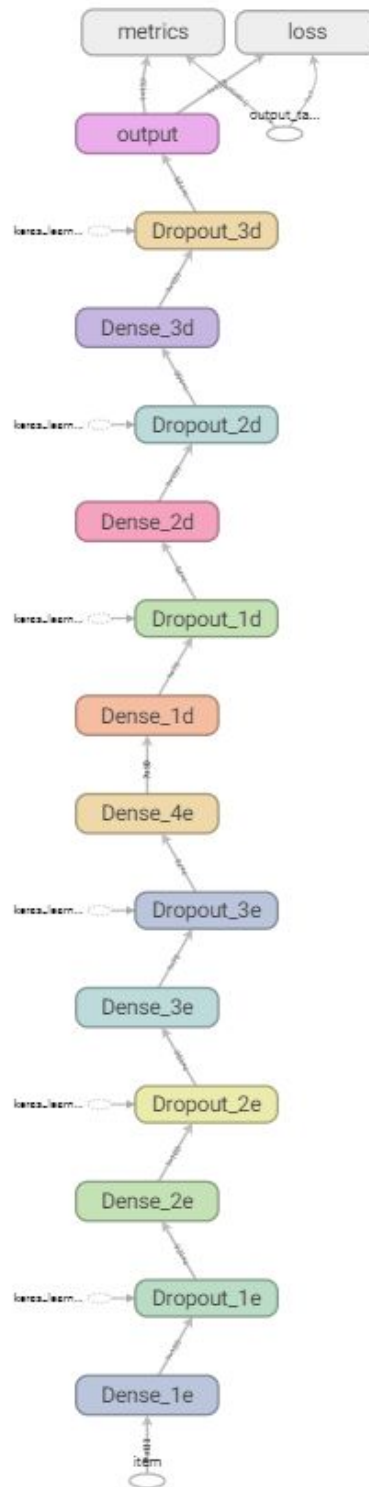


Figure 38: Champion Autoencoder (Content-Based) DNN Model Architecture

With the embeddings created, the next step in the model building process was to leverage the encoded products and identify similar products using cosine and Euclidean distance functions.

The cosine distance function measures the size of the angle between two products in the latent space identifying products that have smallest distance (cosine angle). While this approach identifies similar products, after further inspection, the team realized that it identified products often on the same line or angle, but occasionally these products were actually far away. Figure 39 shows a simple example of how the blue point at (8, 6.4) would have a cosine distance of 1.0 (representing nearly identical products according to cosine distance) whereas the green point (7, 4.2) would have a much smaller cosine distance given it is further off of the line. In latent space, points that are geometrically closer as expressed by Euclidean distance tend to exhibit more similarity than those on the same line or plane.



Figure 39: Cosine Distance Example

Figures 40 and 41 capture a visual representation of the latent product space, where each dot represents a unique product and the color represents the product category. These visuals allow for inspection of the product recommendations, as the seed product is denoted as a star (\*) and the most similar products are marked with a plus (+).

Compared to cosine distance, Euclidean distance measures distance from the two points directly in a radial pattern and identifies products which are close together by proximity. Figure 40 displays a sample product and the top 10 similar products by cosine distance whereas Figure 41 displays a sample product and the top 10 similar products by Euclidean distance.

Through Figures 40 and 41, it is evident that the cosine distance function identified products on an angle from roughly the center down through the green section in the middle to bottom right. In contrast, the Euclidean distance function identified products immediately surrounding the sample product in a circular fashion.

Given the embeddings try to map products by proximity (similar products should be closer together), Euclidean distance proved to be the better option for similar product detection.

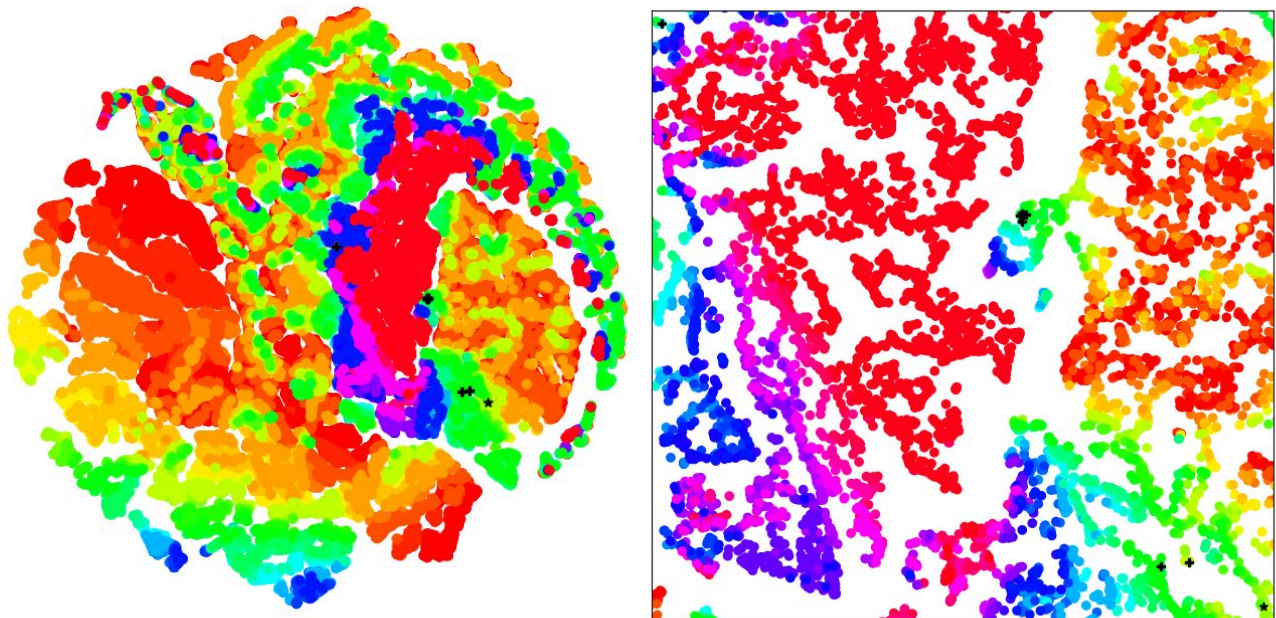


Figure 40: Cosine Similarity of Encoded Products, Colored by Category (Zoomed Right)

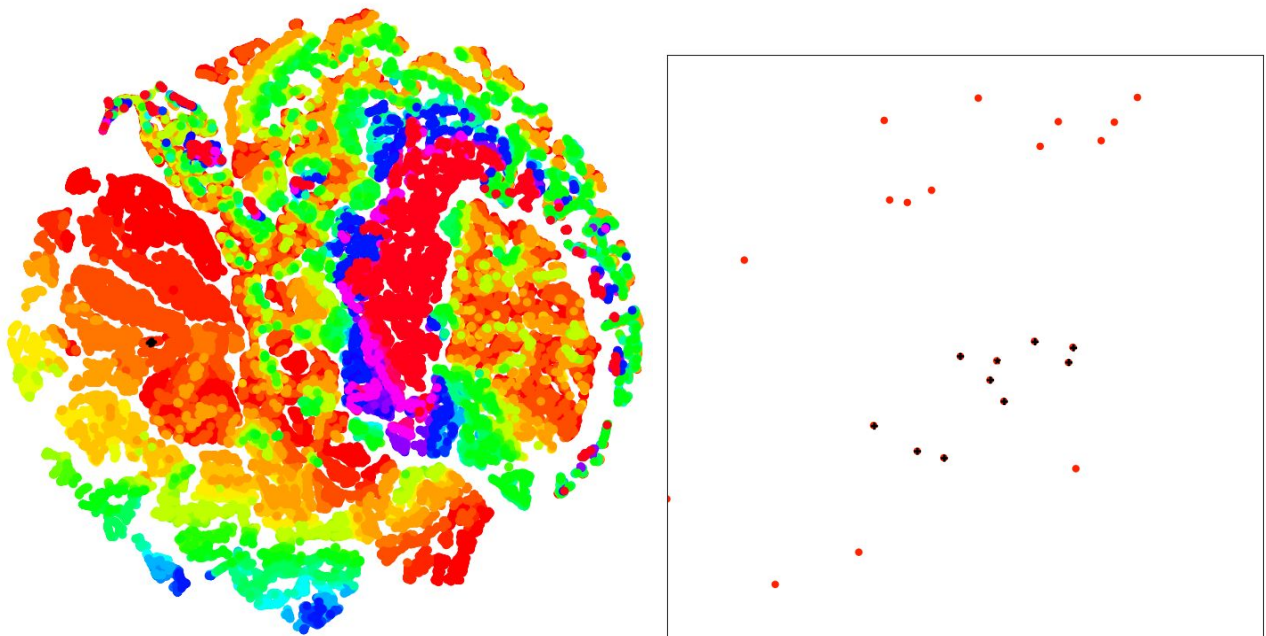


Figure 41: Euclidean Similarity of Encoded Products, Colored by Category (Zoomed Right)

All in all, the autoencoder using Euclidean distance was highly successful in identifying similar products and became the champion content-based model. Figures 42 and 43 display the top 10 similar products for “Black Leather Case/Folio for Apple Ipad With Built-in Stand” ordered by



similarity and distance measure. For this example, the results are very similar between both approaches but they can vary depending on the proximity of products in the latent space, which is where Euclidean distance provided better results.

asin	Distance	Title	Price	Category 2	Category 3
B003EWTM8M	1.000000	Black Leather Case/Folio for Apple Ipad With Built-in Sta	\$ 8.95	Computers & Accessories	Touch Screen Tablet Accessories
B004SMESIQ	0.999798	Griffin CinemaSeat (GB02464) for the&nbsp; iPad (3rd Ge	\$ 8.98	Computers & Accessories	Touch Screen Tablet Accessories
B0054WF1TE	0.999764	SAMSUNG GALAXY TAB 10.1 GEL SKIN CASE - BLACK, WIT	\$ 9.21	Computers & Accessories	Touch Screen Tablet Accessories
B00569Z55A	0.999806	Acase Asus EEE Pad Transformer (TF101) AcaseView Scr	\$ 9.95	Computers & Accessories	Touch Screen Tablet Accessories
B0059KOCNM	0.999851	POSUS Antiglare Antifingerprint Screen Protector for Vi	\$ 9.00	Computers & Accessories	Touch Screen Tablet Accessories
B007505KZO	0.999870	uCase (TM) 360 Degrees Rotating Stand Case for Apple iP	\$ 8.29	Computers & Accessories	Touch Screen Tablet Accessories
B008LBSB06	0.999759	Exact TM Crystal Hard Case Cover for NEWEST Apple Ma	\$ 9.95	Computers & Accessories	Touch Screen Tablet Accessories
B00CHRSAYW	0.999795	Chromo Inc Neoprene Sleeve Case in Light Blue for Chro	\$ 8.95	Computers & Accessories	Touch Screen Tablet Accessories
B00EDICV02	0.999876	Anker Galaxy Tab 3 8.0 Frameless Synthetic Leather Case	\$ 8.99	Computers & Accessories	Touch Screen Tablet Accessories
B00G84C94W	0.999810	Fintie Apple iPad mini 2 with Retina Display Hard Shell Ca	\$ 7.99	Computers & Accessories	Touch Screen Tablet Accessories

Figure 42: Euclidean Distance Similarity Results

asin	Distance	Title	Price	Category 2	Category 3
B003EWTM8M	1.000000	Black Leather Case/Folio for Apple Ipad With Built-in Sta	\$ 8.95	Computers & Accessories	Touch Screen Tablet Accessories
B004SMESIQ	0.999744	Griffin CinemaSeat (GB02464) for the&nbsp; iPad (3rd Ge	\$ 8.98	Computers & Accessories	Touch Screen Tablet Accessories
B0054WF1TE	0.999782	SAMSUNG GALAXY TAB 10.1 GEL SKIN CASE - BLACK, WIT	\$ 9.21	Computers & Accessories	Touch Screen Tablet Accessories
B00569Z55A	0.999494	Acase Asus EEE Pad Transformer (TF101) AcaseView Scr	\$ 9.95	Computers & Accessories	Touch Screen Tablet Accessories
B0059KOCNM	0.999497	POSUS Antiglare Antifingerprint Screen Protector for Vi	\$ 9.00	Computers & Accessories	Touch Screen Tablet Accessories
B005QFH4L2	0.999534	This kit includes: MilitaryShield for your device, installat	\$ 8.98	Computers & Accessories	Touch Screen Tablet Accessories
B007NZUJ4M	0.999750	Manvex Slim and Compact Leather Folio Case Cover for	\$ 9.99	Computers & Accessories	Touch Screen Tablet Accessories
B007505KZO	0.999764	uCase (TM) 360 Degrees Rotating Stand Case for Apple iP	\$ 8.29	Computers & Accessories	Touch Screen Tablet Accessories
B00CHRSAYW	0.999716	Chromo Inc Neoprene Sleeve Case in Light Blue for Chro	\$ 8.95	Computers & Accessories	Touch Screen Tablet Accessories
B00EDICV02	0.999648	Anker Galaxy Tab 3 8.0 Frameless Synthetic Leather Case	\$ 8.99	Computers & Accessories	Touch Screen Tablet Accessories

Figure 43: Cosine Distance Similarity Results

Overall, the CognoClick team was able to achieve more accurate results through a deep learning approach. The team created both types of models (content-based and collaborative filtering models) in an iterative fashion and continuously advanced the models based on learnings along the way. More importantly, the CognoClick team established a foundation for incorporating both metadata and text features into the final product recommendations, which will offer Amazon greater confidence in the personalization of the recommendations returned.

## Ensemble Model

The ensemble model was the last model built by the CognoClick team. The superior performance of the deep learning model allowed CognoClick to deprioritize the ensemble model. Because the ensemble is built on the outcomes of other models, it benefits from the gains and enhancements made to the underlying supporting models. The ensemble was built to serve as a comparison to the hybrid model and ensure that the CognoClick team was selecting the best possible model for Amazon's reviewers.

The ensemble concept enables frequent recommendations across models to be reinforced and further emphasized for end-user consumption. This is similar to a **voting model** in a classification problem. In a voting model, each model generates a prediction regarding a classification. Each model's results are then aggregated and the wisdom of the crowd prevails. Class membership is determined by the consensus of the model. A similar logic was applied to the CognoClick ensemble model for determining the final product recommendations.

The ensemble concept also shares the weaknesses from a processing perspective of the all of the models in the ensemble. The team committed to creating a proof of concept ensemble model. Recognizing this challenge, the ensemble is limited to Clusters 0 and 1 due to the struggles with processing on the baseline model.

The ensemble model was relatively straightforward. Each model's predicted rating was averaged across all models to generate product recommendations for an individual user. If a model did not generate a rating (i.e., there are no ratings for cluster 0 for a particular product) then that model's results were skipped. The results were then sorted by the highest average rating with the top k products recommended to a user. Figure 44 showcases an example of how the averaging would work in the model ensemble.

By averaging the results, it appeared to increase the differentiation in the products being recommended, which is a direct result of the natural variations in the predictions from the models. As a consequence, the ensemble model provided less 5-star ratings and further delineation at the top of the heap for products.

Product	Baseline Rating	Avg. Cluster Rating	Deep Learning Rating	Ensemble
Camera Lens	4.22	4.44	4.70	4.45
Memory Card	4.00	N/A	3.8	3.9

Figure 44: Example of Ensemble Results for a Single Reviewer

Although the CognoClick team piloted the ensemble approach, the results are promising and should be considered by Amazon as a Phase 2 enhancement.

## Dashboard and Mobile Application

While the focus for the POC has centered around building a recommendation engine that generates relevant recommendations, CognoClick still had a goal to provide Amazon's marketing, operations, and analytics teams with an interactive dashboard and mobile application.

Due to the lack of available website data (which captures engagement with recommendations), the team moved away from providing a solution that captured results and operational reporting. Instead, CognoClick created an MVP (Minimum Viable Product) dashboard and mobile application that allows Amazon to better understand what kinds of products are being recommended.

In Phase 2, the team hopes to take advantage of website data to transform the dashboard and mobile application into a tool that allows for measurement of results and business impact. Figure 45 provides high-level screenshots of the dashboard and mobile application, which are described in greater detail in this section.



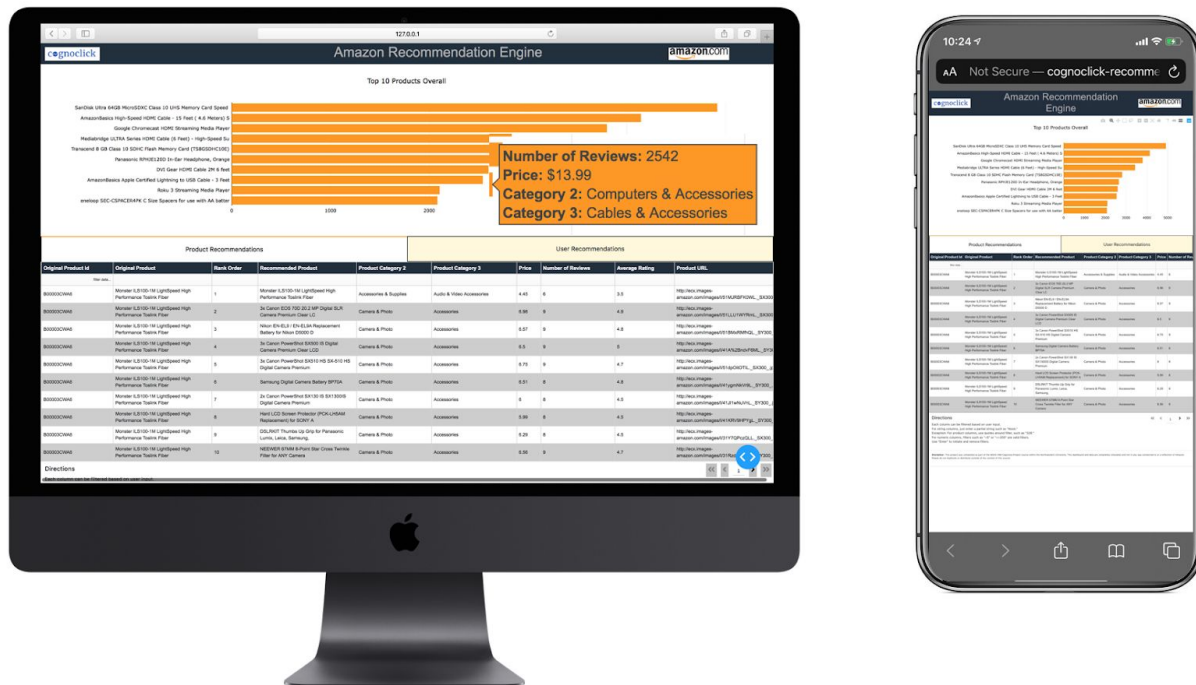


Figure 45: Dashboard (Left) | Mobile Application (Right)

The CognoClick team built the dashboard and mobile application using Dash. Dash is an open-source Python library that enables the creation of reactive web-based applications. The backend codebase simplifies the integration of existing Python code and alleviates the burden of having to code in HTML. The CognoClick team made the decision to use Dash to provide consistency, since all data analysis and models were built using Python. In addition, Dash enabled the team to take advantage of a solution that minimizes development cost and offers full control over customization.

To host the dashboard and mobile application, the team utilized Heroku. Heroku is one of Dash's two preferred options for hosting applications. The other preferred option, Dash's commercial server, was not a viable option for CognoClick to consider for hosting. Amazon will need to evaluate the best hosting option for leveraging this dashboard and mobile application in the future.

Amazon's dashboard and mobile application is divided into two main components:

1. Overall product demand
2. A rank-ordered table containing recommendations for a given product or user

The **overall product demand** visual captures the top 10 most popular products within the electronics product category (see Figure 46 below). This visual provides additional context to the granular recommendations that are shared within the dashboard and mobile application.

Using built-in features from plotly, an open-source library that provides visualization components, the overall demand visual also provides additional information through a tooltip feature. By hovering over each product, Amazon can glean additional context, such as the number of reviews, product price, and product sub categories of a given product. Figure 46 showcases this hover feature.

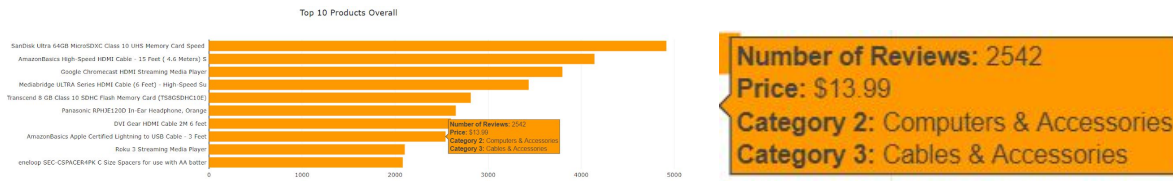


Figure 46: Overall Product Demand (Left) | Hover Feature (Right)

The second and major component of CognoClick's dashboard and mobile application are **two tables at the bottom, which capture recommendations at the product-level and user-level**. These recommendations represent results from the champion deep learning content-based (autoencoder) and collaborative filtering models, respectively.

The product recommendations represent the ten most similar products to a given product (labeled 'Original Product'), while the user recommendations capture the ten best product recommendations that are most relevant for a user. Amazon can easily switch between the different types of recommendations, since the dashboard and mobile application utilize tabs to present this information.

Both tables contain a similar format and capture metadata around the rank-ordered product recommendations, such as the number of reviews, product price, average product rating, and product URL. The tables are both interactive so that Amazon can filter these tables in different ways with content updating dynamically. Moreover, the filters can be combined together, which provides Amazon the ability to understand the product recommendations at a deeper level. Figure 47 illustrates the filtering feature by showing sample filtering on a specific user and their second-ranked product. Noteworthy directions are provided on how to use the filters as well.

Reviewer Id	Rank Order	Recommended Product	F
PSE	=2		
A153667BXYFYPSE	2	Tritton Sound Bite - USB Portable 2.1 Speaker System for PC	C

### Directions

Each column can be filtered based on user input.  
For string columns, just enter a partial string such as "Nook."  
Exception: For product columns, use quotes around filter, such as "328."  
For numeric columns, filters such as "=5" or ">=200" are valid filters.  
Use "Enter" to initiate and remove filters.

Figure 47: Example of Recommendation Filters

Dash was recommended by CognoClick for its extensibility and versatility. The code-intensive Dash application required significant upfront investment to build baseline capabilities. However, the upfront investment affords the application portability across web frameworks. The code base

that was generated to support this initiative can be readily integrated into the existing Amazon enterprise web framework or ported to a new platform or system on the fly.

In addition, the code is flexible and dynamic enough such that it is extensible with continued development as additional data regarding consumer behavior is acquired or considered or additional business needs arise. For instance, Amazon is able to easily incorporate additional key performance metrics (KPIs) or add other tabs to interact with recommendations from multiple models. By leveraging Dash for the dashboard and mobile application, CognoClick had full control over the customization and has left Amazon with a tool that is fully extensible to other product categories and enhancements.

While Dash has shown many benefits, there are some limitations that should be mentioned. First, Dash is not intuitive out of the gate. The CognoClick team had to spend sufficient time understanding how to establish an appropriate layout with reactive components that sufficed for both desktop and mobile views. However, once the team had familiarized itself with the tool, ample time was able to be dedicated toward incremental enhancements.

Second, Dash is a relatively new open-source tool. Given its current state, Dash does not offer a fully mature suite of features. For example, the team wanted to embed hyperlinks within the recommendation tables so that Amazon could view pictures of the recommended products. However, hyperlinks are not supported within tables in the current release of Dash. The lack of hyperlinks was a minor limitation that the team overlooked in light of the extensibility and flexibility offered with Dash. In the end, the team was able to successfully build an MVP solution that allows Amazon to engage with CognoClick's revamped product recommendations.

# Conclusions and Recommendations

## Conclusions

Recommendation engines can be as simple or as complex as a business requires. Much of what is available to an organization is dependent on the quality of the data. In the case of Amazon and their product review data, CognoClick had very high quality data and was able to consider a plethora of models for revamping Amazon's current product recommendations to its reviewers.

Best-in-class recommendation engines utilize a suite of models to generate final product recommendations. Hybrid models or hybrid ensemble models are the preferred industry-standard. Each modeling approach for generating recommendations has its own strengths and weaknesses. By diversifying the models within the recommendation system, sufficient coverage can be provided over the gaps that some models have in their ability to make recommendations. For instance, the cold start problem can be sufficiently remediated by ensuring there are content-based filtering models in the ensemble.

Model validation is difficult without having a user base to test recommendations. The CognoClick team may be able to provide a number of recommendations through the various approaches that have been outlined in this document. However, it is the end user who will be the determinant of the validity of the recommendation. In lieu of having a captive cohort of users for A/B testing or interleaving, the analysis of loss functions and eye tests have had to suffice for this POC. When implementing these models in production, CognoClick recommends leveraging the interleaving method as a means of evaluating and selecting the winning models.

Model building and evaluation were performed in two phases. The first phase (Phase 1 Build) generated outputs that were leveraged for the second phase and gave the team exposure to the potential that each model contained. The second phase (Phase 2 Build) strategically prioritized the development of the models that were most successful in the first wave of model development. The iterative development cycle was immensely valuable to the overall success of evaluating and prioritizing the building of models and ultimately enhanced the final models.

From the first wave of modeling, the baseline model only used product ratings information and generated recommendations leveraging two different approaches. The better of the two approaches was the SVD implementation which produced a MAE of 0.766.

While the baseline model showed initial promise, it posed limitations in trying to generate product recommendations across a full data set. CognoClick utilized Google Cloud Platform (GCP) to try and scale processing, but these attempts were unsuccessful. As a last resort, the team tried generating product recommendations on subsets of data, specifically the user clusters from the review text model, which actually generated results. However, the recommendations that were returned lacked supposed relevancy, and the team decided to not move forward with the baseline model.

To execute on the project goal of including both text and metadata features into a model, the CognoClick team leveraged NLP techniques to prepare review text data for subsequent models. As part of this process, the team also considered a user-user (reviewer) and item-item (product) clustering approach using review text features. Similar products were identified, including a cluster containing all high-quality products. The preliminary analysis from the initial wave of modeling

suggested that clustering appears particularly effective at identifying similar products, more so than similar reviewers.

The clusters built using NLP techniques are an alternative source of information about the product and its quality. Although the actual product rating (1-5) is more direct, it provides less context than what might be hidden in the language. CognoClick was able to extract value from Amazon's unstructured text data. There remains an opportunity to cross analyze these values to find "hidden gems."

CognoClick invested a lot of time and energy into developing various deep learning models in order to successfully implement a hybrid approach. During the first phase of model development, the collaborative filtering deep learning model offered a lift over the baseline model, returning a MAE of 0.7435. The favorable results led to the team pivoting the approach to extend the deep learning model in the second wave of model development.

In the second wave of model development, product metadata and review text features were successfully incorporated into two types of deep learning models. A champion collaborative filtering model was developed using an extensive neural network architecture and generated a MAE value of 0.6620 on the validation set.

Two different types of deep learning content-based models were created from the product embeddings, and one of the models developed was an autoencoder. The autoencoder offered the best results out of any model that the CognoClick team developed and had a MAE of approximately .06. The final product recommendations from both collaborative and content-based deep learning models were inspected, and the content-based model (specifically, the autoencoder) seemed to offer more relevant product recommendations. Both of these deep learning models allow CognoClick to deliver on this project goals of incorporating diverse features and implementing a hybrid approach for its final product recommendations.

To help Amazon's marketing, operations, and analytics teams understand the product recommendations from the deep learning models, CognoClick created an interactive dashboard and mobile application using Dash and Heroku. This dashboard and mobile application allows Amazon to engage with the recommendations and learn more about them. The framework used to develop the dashboard and mobile application is easily extensible as Amazon's needs for understanding its revamped recommendations evolve.

The CognoClick team wrangled large volumes of structured and unstructured data to create a suite of models that all work together to generate powerful and relevant recommendations for Amazon's reviewers.

A summary of CognoClick's conclusions are as follows:

- An agile and incremental model development process allowed the team to continuously advance multiple models and generate insights and learnings along the way.
- Review text features are extremely rich should be leveraged whenever possible.
- A deep learning approach shows vast improvement over a baseline model and offers the ability to incorporate both metadata and text features in an extensible way. If time and

resources allow, a deep learning approach should be considered when building a recommendation engine.

- A hybrid model can utilize the diverse and feature-rich data to a much fuller extent than either content-based and collaborative filtering methods can alone.
- Using review text to create reviewer and product clusters provided the ability to identify similar reviewers and similar products as well as what is important to those reviewers. The initial clustering scheme was more effective at identifying similar products than similar reviewers.
- There is no shortage of opportunity to invest in model development. Amazon will need to leverage a suite of models as this approach offers the best opportunity to generate highly relevant product recommendations.
- Although a model ensemble was considered, it is an inferior option to the deep learning models and a hybrid approach, as it leverages heuristics and is dependent on the validity and constraints of the recommendations of the underlying supporting models.
- Determining the relevancy of product recommendations is difficult without end user feedback. Interleaving is the preferred option to consider when evaluating and selecting final model(s), as end users are able to directly interact with and vote for the winning model(s).
- The electronics data is rich and therefore, computationally expensive. CongoClick leveraged the cloud to take advantage of its computing power and address memory issues. Amazon will need to do the same to support its revamped product recommendations.
- Dash is recommended for engaging with product recommendations and offers both extensibility and versatility.

## Phase 1 Recommendations

The CognoClick team has put together a list of recommendations that Amazon should consider to take full advantage of the revamped electronics product recommendations. These recommendations are broken out into two phases:

1. **Phase 1 (“Building foundation”)**: In this phase, Amazon implements the CognoClick electronics product recommendations and prepares to incorporate enhancements.
2. **Phase 2 (“Enhancing foundation”)**: In this phase, Amazon incorporates enhancements into the CognoClick electronics product recommendations, monitors impact and results, and scales recommendations to other product categories.

Figure 48 captures the Phase 1 recommendations. The list of recommendations for Phase 1 is rather short, since Phase 1 is viewed as a minimum viable product (MVP) for revamped product recommendations. Each recommendation is also described in greater detail below.



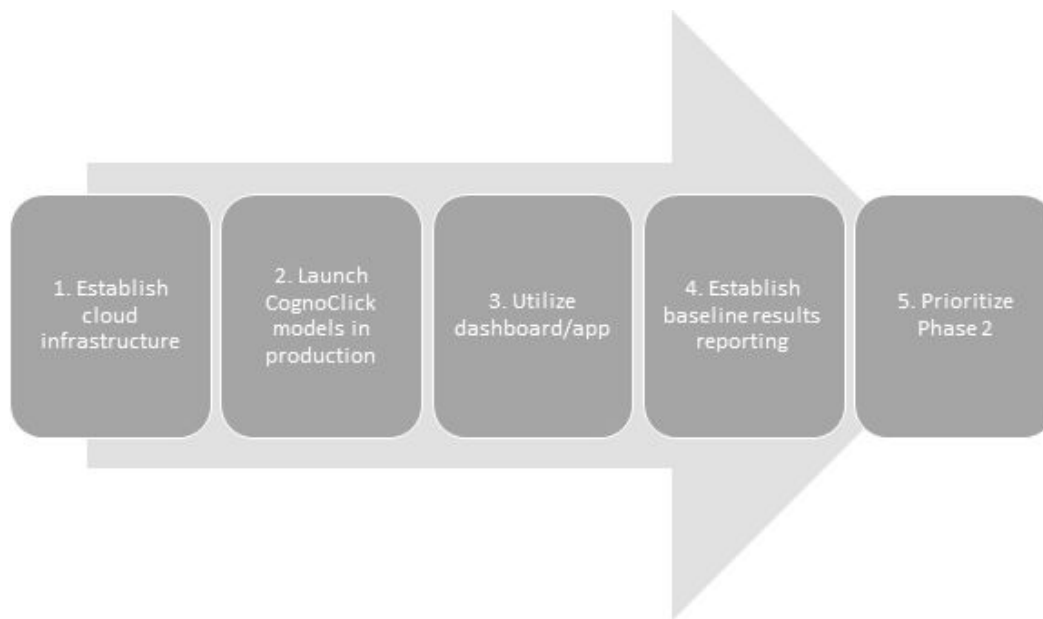


Figure 48: Phase 1 Recommendations

Amazon first needs to **establish a cloud infrastructure** to enable the product recommendations to occur. Since Amazon also offers additional services like Amazon Web Services (AWS), this step should be fairly straightforward.

The CognoClick data processing and modeling phases are computationally expensive, and having a cloud infrastructure will provide the scalability that is needed to move these revamped product recommendations to production. Therefore, this infrastructure will need to integrate with digital properties (amazon.com) and adjust product recommendations within web pages. In addition to hosting and pushing recommendations, this cloud infrastructure will also need access to raw data, such as product metadata and review text, in order to generate new recommendations as user activity occurs and new products become available.

Once the cloud infrastructure is built out, the team will be ready to **launch the CognoClick models in production**. The CognoClick team models can be moved to production as-is. However, Phase 2 identifies some needed enhancements that will ensure these product recommendations remain relevant over time.

With the CognoClick models in production, Amazon's marketing, operations, and analytics teams can **utilize the dashboard and mobile application to understand the recommendations**. The dashboard and mobile application offer exploration into the revamped electronics recommendations. With this information, Amazon can provide context into business results and impact. Phase 2 outlines some key enhancements to better integrate the dashboard and mobile application and enable operational reporting.

In addition to leveraging the dashboard and mobile application to understand recommendations, CognoClick recommends that Amazon **establish some baseline results reporting**. It will be important to understand and communicate the impact that these revamped recommendations are having and more importantly, to assess which enhancements should be prioritized first in Phase 2.



Lastly, Amazon should look ahead and **prioritize Phase 2 enhancements**. With Phase 1 complete, Amazon will have established a foundation for enhancing its product recommendations and gained an understanding of these recommendations. To continue to stay on the cutting-edge, Amazon will need to implement a host of additional enhancements, which will both increase the relevancy of product recommendations and garner more sales.

## Phase 2 Recommendations

To take full advantage of CognoClick's revamped product recommendations, Amazon must implement a series of enhancements to ensure that the recommendations are leading to continuous engagement and bottom-line impacts.

Once Amazon has enhanced its foundation for the electronics product category, it can expand CognoClick's recommendations to other product categories using a similar approach. The model is generalizable and set up such that any other product categories that are passed to the model will generate recommendations. It is advised that Amazon execute the models on a subcategory by subcategory basis to ensure that the increased scale from the additional data does not introduce additional computational complexity into the process.

Figure 49 summarizes the key Phase 2 recommendations. These recommendations should be prioritized according to the investment required and the impact on customer experience. The Phase 2 recommendations are described in further detail below and are listed in no particular order.



Figure 49: Phase 2 Recommendations

The impact of the revamped product recommendations hinges on having an accurate model. Therefore, it will be important for Amazon to **establish a strategy for model retraining**. It takes more resources to retrain a model, so Amazon will have to weigh the cost of retraining the model on a frequent basis (i.e., overnight batch jobs) to the benefits of accurate recommendations. This step is extremely important to not overlook and should be one of the first priorities that Amazon addresses.

Ultimately, to be fully integrated into a web-based consumer platform such as Amazon's website, the recommendations generated by the model will want to occur in near real-time and be natively integrated into the website. Once trained, the model should be fed streaming data with regards to what the potential customer is doing, and based on the activity that the user is performing, return a recommendation.

There are additional enhancements that were being considered by the team that were deferred, including building additional models and revisiting the ensembling approach. The baseline, review text, and deep learning models could all benefit from additional development as well. There will be a point of diminishing returns that would be quickly reached. However, there are opportunities to integrate web-based behavior data and harvest value from the product description text that should be evaluated for potential lift for the recommendation generation.

Amazon's marketing department should play a greater role in assessing the effectiveness of the revamped recommendations. The CognoClick team recommends that Amazon **utilize interleaving, or alternatively A/B testing, to gain feedback on the product recommendations.**

The interleaving approach requires Amazon to shuffle the results of the various models being evaluated and show them all to a user in a randomized order. A customer would be exposed to several models all at once allowing them to pick what they perceive to be the most relevant recommendations. This approach is more common in the recommendation engine industry and is the recommended course of action for Amazon.

In the A/B testing approach, Amazon must feed the revamped recommendations to one group of reviewers and feed a different set of recommendations to another group in order to measure the lift of the revamped recommendations. The control group could be shown the prior recommendations (before CognoClick) or the most popular products within electronics overall.

Regardless of the testing mechanism approach, interleaving or A/B testing will allow Amazon to better understand user preferences and the impact of the recommendations. It will also enable Amazon to fine tune the recommendations to ensure they are having the desired effect.

**Web analytics data** would enhance both the product recommendations and the results reporting. First, it offers the ability to utilize implicit data for product recommendations. The current CognoClick solution leverages explicit data (product ratings) to inform product recommendations. However, with web analytics data, user behavior (i.e., product views and clicks) can be likened to user preferences and thus used to generate even more personalized recommendations. This approach also has the added benefit of being able to scale recommendations to users who have not rated products before, a current limitation in the CognoClick Phase 1 solution.

In addition, incorporating web analytics data into the dashboard and mobile application would allow Amazon to gain a more full picture of the customer experience. Figure 50 shows an illustration of the kinds of insights that could be gleaned from this data. Because this data was not available in Phase 1, a dashboard and mobile application more focused on operational metrics could not be built.

Through the web analytics data, Amazon would be able to directly attribute purchases to the revamped recommendations. This kind of data also opens the doors to inform product placement.

Regardless, the web analytics should be added to the dashboard and mobile application, as it is a key barometer for assessing product performance and impact.

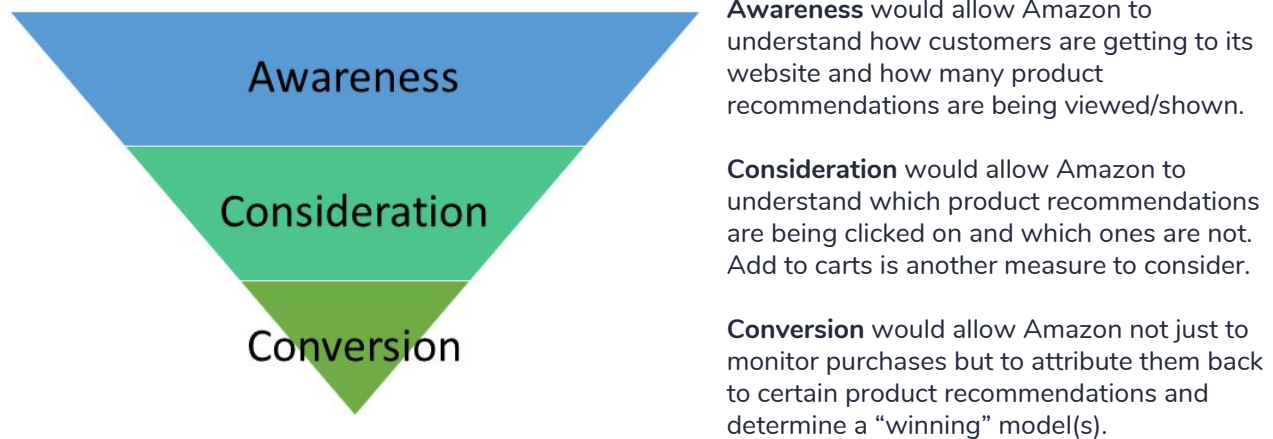


Figure 50: Web Analytics Marketing Funnel

**Enhancing the dashboard and mobile application** is a key consideration in continuing to monitor the results and business impact of the revised product recommendations. The dashboard and mobile application from Phase 1 will allow Amazon to understand the recommendations. However, in Phase 2, Amazon will need to build out a dashboard and mobile application to truly understand results. Defining key metrics of success and adding in other data sources, such as web analytics data, will ensure that the dashboard and mobile application capture the information needed to successfully manage this program.

Once Amazon has enhanced its foundation for the electronics product category, the natural next step is to **expand the revamped recommendations to other product categories**. By enhancing the electronics product foundation first, Amazon will be more equipped to integrate these recommendations into other product categories. CognoClick recognizes the effort required to take this next step and suggests that expansion be managed iteratively in a Phase 3 approach.

In addition to expanding to other product categories, Amazon should consider **expanding these revamped recommendations to all users, not just reviewers**. The CognoClick team had to restrict its scope to reviewers based on the available POC data. However, by starting with a specific population, CognoClick was able to help Amazon develop a strategy that is extensible to other user segments and can be addressed in Phase 3 and beyond.

In addressing CognoClick’s Phase 1 and Phase 2 Recommendations, Amazon will be able to offer more relevant product recommendations to all customers, regardless if they have written a review, and be equipped to understand customer preferences at a deeper level. This information will allow Amazon to drive incremental enhancements to its product recommendations in order to maximize revenue.

## References

Amazon's global career site. (n.d.). Retrieved October 4, 2019, from <https://www.amazon.jobs/en/working/working-amazon>.

AMZN.O - Amazon.com, Inc. Profile. (n.d.). Retrieved from <https://www.reuters.com/companies/AMZN.O>.

Aurisset, J., Ramm, M., & Parks, J. (2017, December 1). Innovating Faster on Personalization Algorithms at Netflix Using Interleaving. Retrieved from <https://medium.com/netflix-techblog/interleaving-in-online-experiments-at-netflix-a04ee392ec55>.

Automated Product Recommendations for All Budgets? (2009, August 27). Retrieved from <http://www.proimpact7.com/ecommerce-blog/automated-product-recommendations-for-all-budgets/>.

BowdenView, J., Bowden, J., & BusinessSmart Ways of Using Google. (n.d.). What A Product Recommendation Engines Mean to Business. Retrieved from <https://www.business2community.com/strategy/product-recommendation-engines-mean-business-0893268>.

Chollet François. (2018). Deep learning with Python. Shelter Island, NY: Manning Publications Co.

Enright, A. (2019, April 25). Amazon's product sales climb nearly 20% in 2018, but only 8% in Q4. Retrieved from <https://www.digitalcommerce360.com/2019/01/31/amazons-q4-sales/>.

Computer Problems. (n.d.). Retrieved from <https://xkcd.com/722/>.

Dash User Guide and Documentation - Dash by Plotly. (n.d.). Retrieved from <https://dash.plot.ly/>.

Economy, P. (2019, August 6). Jeff Bezos Revealed the Secret of Amazon's Stunning Success in Just 3 Words. Retrieved from <https://www.inc.com/peter-economy/jeff-bezos-revealed-secret-of-amazons-stunning-success-in-just-3-words.html>.

Embeddings | Machine Learning Crash Course | Google Developers. (n.d.). Retrieved from <https://developers.google.com/machine-learning/crash-course/embeddings/video-lecture>.

Falk, K. (2019). *Practical recommender systems*. Shelter Island, NY: Manning Publications Company.

Grover, P. (2018, December 18). Various Implementations of Collaborative Filtering. Retrieved from <https://towardsdatascience.com/various-implementations-of-collaborative-filtering-100385c6dfe>.

Hathaway, S. (2015, August 13). Making the Case for Predictive Dynamic Content. Retrieved from <https://www.emailonacid.com/blog/article/email-marketing/making-the-case-for-predictive-dynamic-content/>.

How Many Products Does Amazon Carry? (n.d.). Retrieved from <https://www.retailtouchpoints.com/resources/type/infographics/how-many-products-does-amazon-carry>.

Hug, N. (n.d.). Home. Retrieved from <http://surpriselib.com/>.

Jeane, Presti, M., William, Jay, Hamrick, D., Bugden, D., ... Dana. (2019, June 21). Master the Amazon Sales Rank Top 10 Things you Need to Know. Retrieved from <https://www.junglescout.com/blog/amazon-sales-rank-top-10-things-you-need-to-know/>.

J. McAuley, C. Targett, J. Shi, A. van den Hengel. Image-based recommendations on styles and substitutes. SIGIR, 2015

Kordík, P. (2016, July 12). Recommender systems explained. Retrieved from <https://medium.com/recombee-blog/recommender-systems-explained-d98e8221f468>.

Kordík, P. (2016, August 23). The value of personalized recommendations for your business. Retrieved from <https://medium.com/recombee-blog/the-value-of-personalized-recommendations-for-your-business-6b2e81ce0a4d>.

Lane, H., Howard, C., & Hapke, H. M. (2019). *Natural language processing in action: understanding, analyzing, and generating text with Python*. USA: Manning Publications.

Liao, K. (2018, November 19). Prototyping a Recommender System Step by Step Part 1: KNN Item-Based Collaborative Filtering. Retrieved from <https://towardsdatascience.com/prototyping-a-recommender-system-step-by-step-part-1-knn-item-based-collaborative-filtering-637969614ea>.

Luo, S. (2019, February 6). Introduction to Recommender System. Retrieved from <https://towardsdatascience.com/intro-to-recommender-system-collaborative-filtering-64a238194a26>.

Mcalone, N. (2016, June 14). Why Netflix thinks its personalized recommendation engine is worth \$1 billion per year. Retrieved from <https://www.businessinsider.in/Why-Netflix-thinks-its-personalized-recommendation-engine-is-worth-1-billion-per-year/articleshow/52754724.cms>.

McAuley, J. (n.d.). Real science. Now in real time. Retrieved October 12, 2019, from <https://cseweb.ucsd.edu/~jmcauley/>.

McAuley, J., Pandey, R., & Leskovec, J. (2015). Inferring Networks of Substitutable and Complementary Products. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 15. doi: 10.1145/2783258.2783381

R. He, J. McAuley. Modeling the visual evolution of fashion trends with one-class collaborative filtering. WWW, 2016

Rocca, B. (2019, June 12). Introduction to recommender systems. Retrieved from <https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada>.

Sharma, P. (2019, September 4). Comprehensive Guide to build Recommendation Engine from scratch. Retrieved from <https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-recommendation-engine-python/>.

Vega, N. (2017, March 20). Here's why user reviews on sites like Amazon are such a big deal. Retrieved October 4, 2019, from <https://www.businessinsider.com/amazon-reviews-greatly-impact-online-shopping-sales-2017-3>.

Wu, Z., Tian, H., Zhu, X., & Wang, S. (2018). Optimization Matrix Factorization Recommendation Algorithm Based on Rating Centrality. Data Mining and Big Data Lecture Notes in Computer Science, 114–125. doi: 10.1007/978-3-319-93803-5\_11

Welcome to Surprise' documentation!. (n.d.). Retrieved from <https://surprise.readthedocs.io/en/stable/index.html>.

Zucconi, Alan. (2019, August 16). An Introduction to Neural Networks and Autoencoders. Retrieved from <https://www.alanzucconi.com/2018/03/14/an-introduction-to-autoencoders/>.

## Appendix

### Code

The team used Github to store and manage all code for the project.

Here is a link to the repository: <https://github.com/dashpound/capstone>.

## Dashboard and Mobile Application

The team created a separate repository to deploy the dashboard and mobile application via Heroku.





Here is a link to the final dashboard and mobile application:  
<https://cognoclick-recommendations.herokuapp.com/>.

Here is a link to the separate Github repository:  
[https://github.com/dashpound/review\\_dashboard](https://github.com/dashpound/review_dashboard).

## Project Team

CognoClick was born out of four Northwestern University MSDS students who have a passion for shopping, machine learning, and natural language processing. The team has strong technical and communication skills and is dedicated to delivering above and beyond project goals. Each team member also brings unique interdisciplinary skills that will help keep the project moving forward at all times. The project roles are summarized in the table below.



P = Primary S = Secondary * = All participating	<b>John Kiley</b> 	<b>Brian Merrill</b> 	<b>Hemant Patel</b> 	<b>Julia Rodd</b> 
Project Manager	S			P
Technical Writer	S	S	S	P
Programmer	P	S	P	S
Dashboard/Mobile App		P	S	
Oral Presentation	*	*	*	*

**John Kiley** is a Director of Data & Analytics at a financial services company and has a record of tackling challenging business problems and working in turn-around organizations. John's primary focus is on enabling better business outcomes through data driven decision making. John has several years experience leading machine learning, system conversion, data warehousing, and business intelligence projects. John will serve the role of programmer and will support project management and technical writing activities of the project.

**Brian Merrill** is a Managing Director at a Big 4 consulting firm providing data analytics services to clients to help solve their complex legal and compliance issues. With over 15 years experience developing applications, analyzing systems, data mining and performing data analysis, Brian brings significant insight and extensive client experience to CognoClick to help solve the difficult challenges of product filtering and recommendation. Brian will serve as the primary visualization expert as well as supporting the technical writing and development of the recommendation engine.

**Hemant Patel** is a Senior Manager at a performance marketing agency responsible for designing, developing, and implementing predictive modeling solutions for clients looking to optimize their marketing strategies. Hemant has an extensive background that includes experience in data mining, predictive modeling, and business analytics. Hemant will serve as a programmer as well as support technical writing and visualization efforts.

**Julia Rodd** is a Senior Data Scientist at a financial services company and has several years' experience leading and participating in analytics projects. She brings an ability to think strategically, technical knowhow, and a focus on driving for results. Julia will be serving the role of project manager and primary technical writer. She will also support the development work as needed.



## Table 1 - Electronics Data Overview

Below are tables outlining the column names, definitions, and example data for the three datasets used in the POC.

### Reviews Data

Field Name	Definition	Example Data
reviewerID	The ID of the reviewer	A2JXAZZI9PHK9Z
asin	The ID of the product. Can be used to join to other data sets.	0594451647
reviewerName	The name of the reviewer.	Billy G. Noland "Bill Noland"
helpful	The helpfulness rating of the review.	[3, 3]
reviewText	The text of the review.	I am using this with a Nook HD+. It works as described. The HD picture on my Samsung 52" TV is excellent.
overall	The overall star rating of the product.	5
summary	The summary (caption) of the review.	HDMI Nook adapter cable
unixReviewTime	The time of the review in unix time.	1388707200
reviewTime	The time of the review in raw format.	01 3, 2014

### Product Metadata

Field Name	Definition	Example Data
asin	The ID of the product. Can be used to join to other data sets.	0594451647

title	The name of the product.	Barnes & Noble HDTV Adapter Kit for NOOK HD and NOOK HD+
price	The price in US dollars at the time of data capture (circa August 2014).	49.95
imUrl	The URL of the product image.	http://ecx.images-amazon.com/images/I/51RjSETO23L._SX300_.jpg
description	The description of the product.	HDTV Adapter Kit for NOOK HD and NOOK HD+\nThi...
related	The IDs of related products. Distinguishes between also bought, also viewed, bought together, and bought after viewing.	{'also_bought': ['B009L7EEZA', 'B00AGAYQEU', 'B009L7EJAK', 'B00C2L6MAW', 'B00BN1Q5ZE', 'B009QZH7BU'], 'bought_together': ['B009L7EEZA'], 'buy_after_viewing': ['0594481813', '0594481902', 'B009L7EEZA', 'B00AK2MHEU']}
salesRank	Sales rank information.	nan
brand	The brand name of the product.	nan
categories	The list of categories that the product belongs to.	[['Electronics', 'Computers & Accessories', 'Touch Screen Tablet Accessories', 'Chargers & Adapters']]

## Question and Answer Data

Field Name	Definition	Example Data
questionType	The type of question. Options are yes/no or open-ended.	yes/no

asin	The ID of the product. Can be used to join to other data sets.	0594033926
answerTime	The time of the answer in raw format. Some formats are relative while others are exact dates.	17 days ago
unixTime	The time of the answer in unix time.	NaN
question	The text of the question.	Will this fit a Nook Color that's 5 x 8?
answerType	The type of answer, such as 'Y' or 'N.'	Y
answer	The text of the answer.	yes

**Table 2 - Reviewer Features**

Field Name	Definition	Example Data
reviewerID	The ID of the reviewer	A000715434M800HL CENK9
MaxRating	Maximum rating value for each reviewer	5
MinRating	Minimum rating value for each reviewer	1
NumberOfRatings	Total number of ratings for each reviewer	5
AverageRating	Average rating value for each reviewer	3.2
MedianRating	Median rating value for each reviewer	3
SummedRatings	Sum of all rating values for each reviewer	16
MaxPrice	Maximum product price for each reviewer	95.18
MinPrice	Minimum product price for each reviewer	11.99
AveragePrice	Average product price for each	46.11

	reviewer	
MedianPrice	Median product price for each reviewer	25.99
SummedPrice	Sum or price across all reviews for each reviewer	230.55
SummedHelpfulNumer	Sum of up-votes for each reviewer	13
SummedHelpfulDenom	Sum of total-votes for each reviewer	20
MaxNumDaysBetweenReviews	Maximum number of days between reviews completed for each reviewer	85
MinNumDaysBetweenReviews	Minimum number of days between reviews completed for each reviewer	0
AverageNumDaysBetweenReviews	Average number of days between reviews completed for each reviewer	35
MedianNumDaysBetweenReviews	Median number of days between reviews completed for each reviewer	30
SummedNumDaysBetweenReviews	Sum or total days between each sequential review completed	360
helpful_flag	Binary flag indicating if helpful votes were completed by each reviewer	1
helpful_proportion	Ratio of up-votes relative to total-votes for each reviewer	0.67

**Table 3 - Product Features**

Field Name	Definition	Example Data
asin	The ID of the product. Can be used to join to other data sets.	0528881469
description	The description of the	Like its award-winning

	product.	predecessor, the Intell...
title	The name of the product.	Rand McNally 528881469 7-inch Intelliroute TND...
category2_t	Amazon's second-level category for a product.	GPS & Navigation
hasDescription	A binary variable indicating if a product has a product description.	1
price_t	The price in US dollars at the time of data capture (circa August 2014).	299.99
containsAnySalesRank	A binary variable indicating if a product has any sales rank.	0
numberQuestions	The number of questions/answers for a given product.	0
numberReviews	The total number of reviews written for a given product.	5
meanStarRating	The average star rating across all reviews for a given product.	2.4
star1Rating	The proportion of reviews with a 1-star rating.	.4
star2Rating	The proportion of reviews with a 2-star rating.	.2
star3Rating	The proportion of reviews with a 3-star rating.	.2
star4Rating	The proportion of reviews with a 4-star rating.	0
star5Rating	The proportion of reviews with a 5-star rating.	.2

## Table 4 - Baseline Model Cross Validation Results

	Fold 1	Fold 2	Fold 3	Mean	Standard Deviation
RMSE	1.034	1.033	1.030	1.032	.002
MAE	.775	.775	.772	.774	.001
FCP	.553	.547	.560	.553	.005
Avg precision at k	.923	.921	.922	.922	.001
Avg recall at k	.545	.551	.551	.549	.003

SVD 3-Fold Cross Validation Results

	Fold 1	Fold 2	Fold 3	Mean	Standard Deviation
RMSE	1.090	1.089	1.087	1.089	.001
MAE	.807	.808	.805	.806	.001
FCP	.582	.575	.571	.576	.005
Avg precision at k	.886	.885	.887	.886	.001
Avg recall at k	.554	.555	.558	.556	.002

KNN 3-Fold Cross Validation Results

## Figures 1 and 2 - Reviewer Text Clusters

The distribution of reviewers in each cluster for all electronics data is outlined in Figure 1 below. The largest cluster is Cluster 5, which contains 22.7 percent of all reviewers, followed by Cluster 4 with 17.3 percent of all reviewers. The smallest cluster is Cluster 7, which contains 5.5 percent of all reviewers.

Label	Cluster 0:	Cluster 1:	Cluster 2:	Cluster 3:	Cluster 4:	Cluster 5:	Cluster 6:	Cluster 7:	Cluster 8:	Cluster 9:
Members	13,995	20,058	18,719	13,866	33,263	43,717	14,480	10,591	11,570	12,144

Figure 1: Distribution of reviewers in clusters (Full electronics data)

Taking a further look into each cluster, the top 10 key terms are identified in Figure 2 below.

Label	Cluster 0:	Cluster 1:	Cluster 2:	Cluster 3:	Cluster 4:	Cluster 5:	Cluster 6:	Cluster 7:	Cluster 8:	Cluster 9:
1	ipad	sound	camera	cable	great	one	drive	lens	unit	mouse
2	case	headphones	use	great	product	like	usb	camera	one	keyboard
3	great	speakers	great	works	works	use	one	great	use	use
4	one	good	one	one	good	get	hard	use	would	one
5	keyboard	quality	good	use	price	would	use	one	good	great
6	like	great	battery	good	well	good	great	good	great	like
7	use	like	would	well	use	dont	would	well	well	good
8	screen	one	like	quality	would	great	works	get	get	would
9	would	use	well	would	recommend	well	computer	like	works	works
10	well	music	get	usb	one	really	good	would	like	usb

Figure 2: Key terms identified by reviewer clusters (Full electronics data)

Interestingly enough, Clusters 4 and 5 appear not to be purchasers of any particular products and seem to be particularly positive reviewers. The top terms across these two clusters all contain terms that carry a generally positive sentiment with the exception of the word “don’t” in Cluster 5.