CAN CLASSIFICATION PLACE NEW BOOKS?

Predicting the placement of Seattle Public Library books based on collections data

Abstract

INFO634-900: This project applies the machine learning Random Forest algorithm to a dataset of frequent checkout materials at the Seattle Public Library to simulate prediction of new item placement within the Library's holdings. It accomplishes this by training on groupable features like Subjects, Authors, and Item Collections in 2/3 of a frequent checkout set to predict physical locations of the 1/3 testing set. Accuracy for the final model is poor, averaging near 51%. The project concludes that the Seattle Public Library's storage logic does not fit an 'each place is unique' approach, rather that satellite locations have very similar collections which makes them difficult to distinguish between using machine learning.

Charles Zange

crz34@drexel.edu

Sunday, June 9, 2019

Contents

1. Introduction

1.1 Problem description and scope of work

"Now, where should I put that book?"

Some libraries steward millions of items across a broad range of formats. Librarians rely on databases to keep everything in order, especially the location of current items. But what about the placement of new items? Where should the *next* book go, the book that the library just purchased to expand its holdings? Is there a logic that defines where items should be placed – and is that logic interpretable using data mining?

In the spirit of this project, there is an implied supply-and-demand incentive structure for expanding the library's holdings. Items that are checked out would motivate the acquisition of new items for similar subjects, media, and collections. Therefore, in absence of a specific dataset of only new books, this project will draw from active collections frequently checked out by library patrons and build an algorithm to explore how a library places books and use a 2/3 + 1/3 split to simulate a 2/3 collection of materials being augmented by the 1/3 set of 'new' books.

In summary, this project will evaluate a sample of the Seattle Public Library's (SPL) collection data using a classification algorithm to predict where a checkout item is physically located in SPL's collection based on other features. The classification algorithm will utilize a set of features that are potentially common with new acquisitions (things like 'Subject', 'Collection', and 'ItemType') rather than unique to individual collection items (like 'Title' or 'ISBN'). The goal of this project is to develop an algorithm to assist with classifying newly acquired books and media into a storage location.

1.2 The Seattle Public Library

The Seattle Public Library System unites 27 public learning spaces throughout the city of Seattle. In this project, this 27-member system will be referred to collectively as the Seattle Public Library (SPL). SPL opened in the late nineteenth century and has expanded under the direction of their Chief Librarian, a

position currently occupied by Marcellus Turner [2019]. In recent years – and no doubt with support from Turner – SPL has expanded its online profile through *data.seattle.gov* with datasets on the overall holdings, checkouts by title, and more. SPL's data also came to the online distribution platform Kaggle [City of Seattle 2019a, 2019b, Seattle Public Library 2017]. These datasets are often maintained by Kaggle directly (as in the example of the Integrated Library System Dictionary) and receive periodic updates [City of Seattle 2019e]. The largest of these collections is the Seattle Library Collection Inventory, 26.8 million rows of observations straight from SPL's total holdings [City of Seattle 2019b]. This dataset, while expansive and exciting, contains millions of books and media that have rarely if ever been checked out by the public (more on this below). This project will instead focus on a dataset of SPL's checkout records from 2005 to 2017, a 2.69 million observation subset of SPL's holdings comprised only of those physical items that have been checked out by library card holders in that twelve-year period [Seattle Public Library 2017]. There are several advantages to using the smaller checkout records dataset:

1. Checkout reflects the active collection: Of the 26.8 million items in SPL's reported holdings on Kaggle [City of Seattle 2019b], only 2.69 million physical items were checked out by card holders in the 2005-2007 period. This reflects about 10% of the total collection. The other 90% of the collection is more complex, containing both low-interest items that are rarely touched and restricted items, like reference collections, that are not available for checkout. This more closely matches the mock 2/3 collection vs. 1/3 new materials approach taken in this project.
2. Physical items-only is necessary for location prediction: The checkout dataset for this project includes only physical checkout items like books and DVDs, and not digital items from SPL's online collections. This is critical for this project, which targets physical storage locations.
3. The dataset is stable, last updated 2017: Both SPL and Kaggle provide up-to-date datasets through *data.seattle.gov* and the Kaggle website. For this first project, a stable dataset was

selected to simplify the experiment. This guaranteed consistent data for the 2/3 vs. 1/3 splitting rather than developing an algorithm on a moving target.

1.3 Working with cultural heritage data

My background is in museum object care and digitization. I have worked in a variety of archival and museum settings, building up knowledge in object management in several cultural heritage organizations. Based on jobs I have performed and subsequent research on organizational policies, here are an understanding, a reservation, and an insight that I brought to this project.

1. An understanding: cultural object acquisition is not a scientific process. It is a human process. In museums, objects are first acquired through a transfer of title (purchase, donation, etc.) and then accessioned into a specific collection through a second process of adjudication [Buck Gilmore eds. 2010]. Individual curators, various committees, strategic plans, executive directors, and available funds dictate which items to accession. With this project, I hoped to narrow the focus away from the intangible selection of *which* books to buy and toward the more tangible process of placing material into physical spaces.
2. A reservation: not all storage spaces are created equal. The dataset for this approach has the limitation of treating each 'ItemLocation' (the target classifying feature) outcome as functionally equivalent. An algorithm based from this dataset cannot distinguish whether a specific 'ItemLocation' is available for storage. This goes beyond the capacity of the space: certain material types have specific storage needs that a target space may not have. Differences in relative humidity, temperature, and storage materials like acid-free or buffered papers have different applications in storage environments [Buck Gilmore eds. 2010]. Improving the algorithm would require additional data about the specific facilities that safeguard SPL's holdings. By including 'ItemCollection' within the training environment, I hoped to improve the

overall accuracy of the algorithm, as collections are frequently grouped within the same storage area within cultural organizations like libraries.

3. An insight: cultural object placement is a logical process. If we take a market-based theoretical approach to this problem, books should ultimately find an optimal storage location based on some logical determination of 'like with like', 'as space allows,' etc. What I mean by this is that the placement of books should reject the null hypothesis in statistics. SPL's method for placing new books should not be random. My experience in museums has shown that item placement is performed carefully and rationally given many considerations and competing circumstances. That said, it remains to be seen if cultural object placement is a *replicable* process. The logic that guided one individual to place a book in Storage A may instead guide another person to place the book in Storage B. This downstream randomness in the decision tree within an organization like SPL will have an impact on a machine learning algorithm's overall accuracy.

1.4 Supporting projects

Two dataset kernels by developers on Kaggle introduce exploratory data analysis on SPL's checkout data. Both kernels were consulted during the initial exploration of this dataset.

The first kernel, "Library Checkouts EDA," explores the checkouts by title dataset with support of a method that changes data types in the dataset to reduce memory usage. The memory usage method was not necessary for this project, but the method could have application when considering the larger 26.8 million observation dataset of the total SPL holdings. "Library Checkouts EDA" also provided guidance on working with the checkouts metadata and FAQs page to make sense of the encoded variables within the dataset (see section 2 below). This provided important insight during the initial exploration of the data for this project [Ofer 2019].

The second kernel, "Review of Seattle Library Checkout Records," ties in the collection inventory with a list of checkouts and the data dictionary provided by SPL. In so doing, this kernel ties together

measurements in circulation over time with the item titles to look for frequently checked-out materials. This overview of the data helped in understanding the relationships between and frequencies of specific item checkouts and overall trends in SPL's frequently accessed collections. These insights would be helpful for SPL when considering new items to add based on popular subjects, which in turn would feed into this project's algorithm on the physical placement of new items into storage. While this project is limited to just the inventory data and not frequencies of circulation, there is ample opportunity to expand the algorithm of this project using these insights [Murphy 2019].

2. Data Descriptions

2.1 Data exploration: descriptions, counts, and feature selection

In all, the dataset is 2.69 million observations with thirteen features. Appendix A contains a full summary of each feature. Below are tables for features employed by this project. The Description within each feature table is copied directly from SPL's own data dictionary, hence the use of italics [City of Seattle 2019e].

Subjects	
Type: Nominal, multi-value	
Description: *A comma-separated list of the subject authority records associated with the title, including Motion Pictures, Computer Programming, etc. Typically these are highly specific*.	
Count: 2621275	Unique: 439811
Most frequent value: Rock music 2011 2020 (10945 count)	
Note: The total count of this feature is misleading. The fourth most frequent value is "Popular music 2011 2020, Rock music 2011 2020" with 5033 occurrences. This demonstrates that subjects are variably listed, from zero or one to several or more in a single item. There are items where "Subject A, Subject B" are not alphabetized, leading to double-counting of "Subject B, Subject A". However, alphabetizing will change the order of subjects which may, in some collections, represent a hierarchy with primary subjects coming before others. For this reason, Subjects were not alphabetized or separated for testing.	

--

ItemType	
Type: Nominal, encoded, single-value	
Description: *Horizon item type. Look up value descriptions in HorizonCodes using CodeType "ItemType".*	
Count: 2687149	Unique: 63

Most frequent value: Book: Adult/YA (1338207 count)
Note: This is an encoded feature. In the source dataset, ItemType is listed in codes like “acbk” and “jcbk,” presumably to save on the storage footprint of the total collection. These are somewhat interpretable from first sight (“acbk” ➔ adult collection of books ➔ “Book: Adult/YA”), however the data dictionary makes substitution straightforward for use in the dataset. Each observation belongs to only one ItemType, with “Book: Adult/YA” greatly outnumbering the rest.

--

ItemCollection	
Type: Nominal, encoded, single-value	
Description: *Collection code for this item. Look up value descriptions in: Integrated Library System (ILS) Data Dictionary.*	
Count: 2687149	Unique: 193
Most frequent value: CA-Reference (384170 count)	
Note: This is an encoded feature. In the source dataset, ItemCollection is listed in codes like “canf” and “nanf” presumably to save on the storage footprint of the total collection. These are somewhat interpretable from first sight (“canf” ➔ “CA Reference”), however the data dictionary makes substitution straightforward for use in the dataset. Each observation belongs to only one ItemCollection.	

--

FloatingItem	
Type: Boolean, single-value	
Description: *Label that indicates if an item floats.*	
Count: 2687149	Unique: 2
Most frequent value: Floating (411108 count) (other 2276041 are nulls)	
Note: An item that “floats” is moved periodically from one storage location to another, according to the data dictionary. These items make the librarian’s lives potentially easier as books increasing in popularity can be kept closer to the main facility, however it presents a significant challenge for a machine learning algorithm trying to classify a specific location. For this project, FloatingItems are not included in the analysis. Also note: removing FloatingItems from the analysis caused three ItemLocations to drop, bringing the total number of predictable ItemLocations to 30.	

--

ItemLocation	
Type: Nominal, encoded, single-value	
Description: *Location that owned the item at the time of snapshot. 3-letter code. Note: as of 2017, some items are "FLOATING" which means they don't necessarily belong to a specific branch. Location of given copy could change based on where the item is returned.*	
Count: 2687149	Unique: 33
Most frequent value: Central Library, 1000 4TH AV (1080480 count)	
Note: This is an encoded feature. In the source dataset, ItemLocation is listed in codes like “cen” and “net” presumably to save on the storage footprint of the total collection. These are somewhat interpretable from first sight (“cen” ➔ “Central Library”), however the data dictionary makes substitution straightforward for use in the dataset. Each observation belongs to only one ItemLocation.	

Table 1 - Features used for testing.

Below is the total feature space with qualifying and non-qualifying fields, plus minor descriptions on the selection process.

Qualified?	Feature	Why the feature did/did not qualify
Qualified	Author	Authors shared between books may impact placement.
	PublicationYear	Books of the same age may be placed together, especially in historical collections.
	Publisher	Historical publishers that are no longer active may help clue in the algorithm on the placement of books.
	Subjects	Multi-item feature that establishes 'like with like' matching.
	ItemType	Media type and adult vs. children audiences affect storage needs, impacting the algorithm.
	ItemCollection	Distinctions between Reference and non-Reference collections impact the algorithm.
	ItemLocation	Target class.
	FloatingItem	Removes items with multiple locations from consideration. Used for filtering, but not included in the final model
Did not qualify	BibNum	Specific on a per-item basis, does not fit spirit of exercise.
	Title	While a Title can be shared across multiple media (books, films, etc.), the Title feature was ultimately dropped to reduce the per-item specificity of the analysis.
	ISBN	Specific on a per-item basis, does not fit spirit of exercise.
	ReportDate	This date refers to when the data were pulled from the database and not specifically to the items in SPL's holdings.
	ItemCount	The count of items per line did not fit into this testing model.

Table 2 - Full feature list with decisions on qualification.

2.2 Data exploration: Accuracy Score

After an initial triage of the features for testing, I evaluated said features to discover which field's variance was highest relative to the variance of classifications. I employed the same method here as in a previous machine learning experiment predicting video game match outcomes based on in-game data: the ANOVA F-test. This calculation, in broad terms, evaluates the variance within a feature against the variation of classifications within the same observations as that feature. Python's Scikit library has a tool to perform this kind of testing called SelectKBest in which I applied the *f_classif* module to perform an ANOVA F-test. Python delivers its outcomes in an accuracy score rather than a pure F-score. The accuracy score is a relative measure of variation by classification variance: the higher the accuracy score, the better the feature is at classifying the outcome. In this project, a high Scikit accuracy score means

that the feature is stronger at predicting where a book will be placed than its counterparts [Scikit-learn 2019, Minitab 2016]. The results of this evaluation are shown below:

Feature	'Author'	'PublicationYear'	'Publisher'	'Subjects'	'ItemType'	'ItemCollection'
Accuracy score	3600.1039	302.0697	1659.1432	3518.2032	12.0839	43.4779

Table 3 - Accuracy scores of tested features.

From these measures, we observe that 'Author' has the highest variance relative to the total variance of observations, followed by 'Subjects' and 'Publisher.' In theory, this means that 'Author' changes more frequently as the outcome, 'ItemLocation,' changes relative to other features. From this initial testing, I would estimate that 'Author' is the most discriminating feature among those selected for testing. 'ItemType', conversely, is the least discriminating feature, and I would expect that it will not weigh heavily on the algorithm's predictions.

2.3 Target feature: 'ItemLocation'

There are 29 'ItemLocation' options within the dataset. In the initial dataset, each 'ItemLocation' is given a 3-letter encoding. I decoded the itemset, and grouped the top 10 frequent values within 'ItemLocation' below (Figure 1). The colors of this visualization correspond to the average item count per observation, understood to be the number of copies of a single observation within the 'ItemLocation'. The Central Library has both the largest storage of SPL's holdings and the highest average ItemCount. It holds the most books as well as the greatest percentage of books with multiple copies relative to the other top 10 'ItemLocation' options.

The largest 'ItemLocation' is encoded as 'cen', meaning the Central Library on 1000 4th Avenue. 'cen' accounts for 57.9% of the total holdings of SPL across every collection. The remaining 28 'ItemLocations' house 42.1% of the total holdings. These 28 smaller locations are spread around Seattle at SPL satellite libraries or offsite storage facilities. There is also one error in the encoding detected during preprocessing. The 'mob' three-digit code does not correspond to an active location in SPL's data

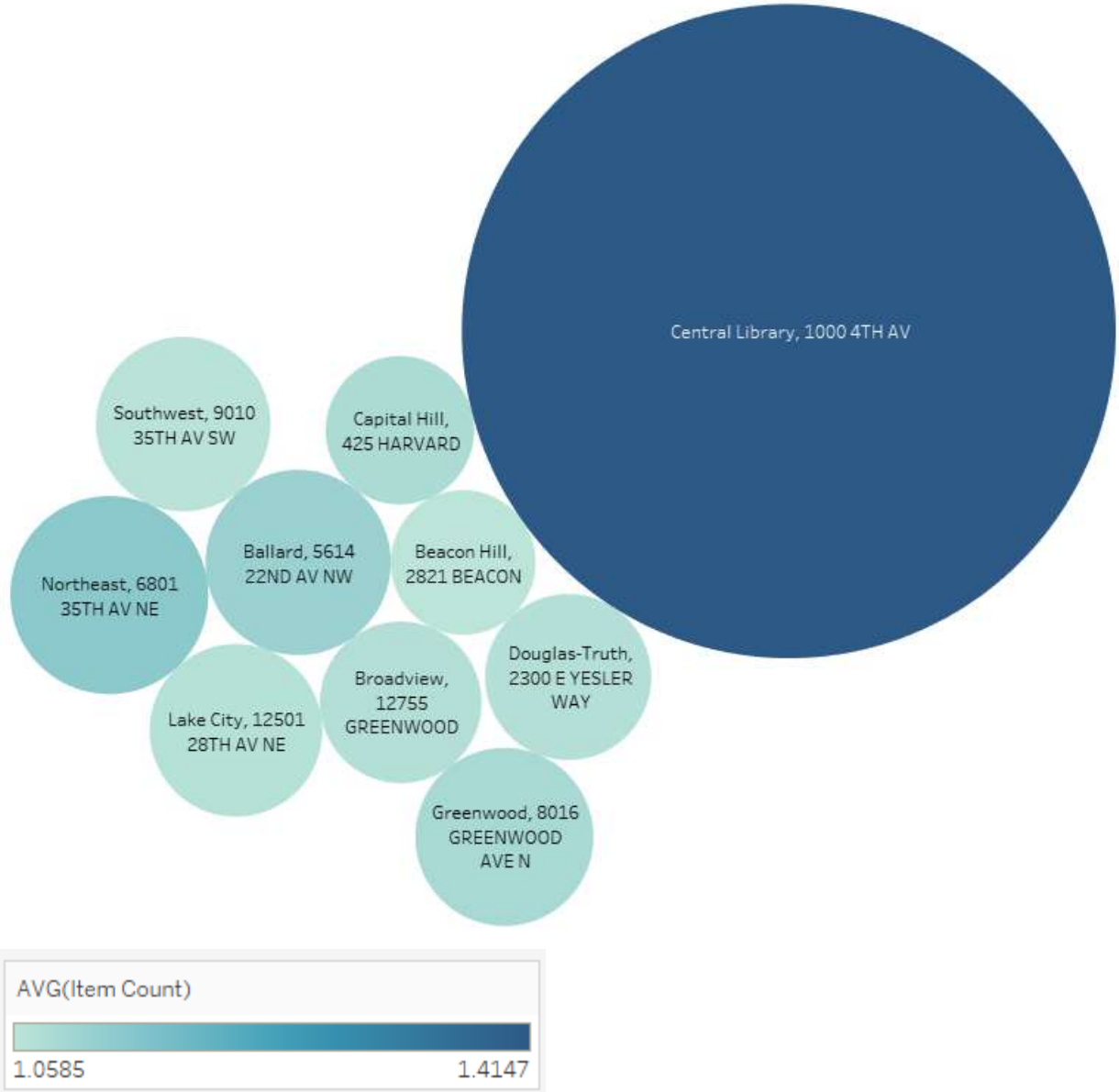

Figure 1 - Top 10 storage locations by number of observations, with color chart for average item count per observation.

dictionary [City of Seattle 2019e]. 'mob' may be a location that is excluded from SPL's data dictionary (perhaps for reasons of security?), it may reflect a location no longer in use (more likely), or it may be an error within the data entry that has not yet been corrected (did the staff member mean 'mon' for Montlake, 2401 24TH Ave?). In my research I could not confirm if 'mob' was an intentional location or an error. Ultimately, I left the 9,363 'mob' observations in the dataset as a precaution.

3. Methodology

3.1 Resources

This project drew from the following resources:

Python 3	All pre-processing and machine learning was written in this programming language. Python was also used to generate confusion matrices.
Scikit	Python's scikit.learn library provided the machine learning tools.
NumPy	Python's mathematical library performed vectorized operations and prepared data for the Random Forest algorithm.
Pandas	SPL's data were loaded into Pandas dataframes for processing.
Jupyter Lab	Jupyter Lab, built from the Anaconda distribution of Python 3, provided the coding environment.
Tableau	Tableau was used to create some of the visualizations.

Table 4 - Software used in this project.

This project was coded on a Windows 10 Pro operating system, but the Python 3 code is portable to other platforms as well.

3.2 Approach

The data were first preprocessed to eliminate nulls and unnecessary features. The resulting clean dataset was then split into 2/3 training and 1/3 testing sets for the Random Forest classifier. Results were displayed in Python using an accuracy score and a confusion matrix. A second confusion matrix based solely on 'ItemLocation' values outside of the central library was added to provide greater context on the dynamics of the data. There are four factors that impacted or shifted the methodology during testing.

1. Why was a Random Forest employed over other algorithms? I selected a Random Forest for three reasons: (A) Random Forests handled categorical, multi-class data natively; (B) These models tend to train quickly relative to Artificial Neural Networks, making them ideal for repeated testing during data discovery; (C) In comparing the performance of other algorithms (including artificial Neural Networks), I found that Random Forest performed as well if not better on overall accuracy than its competitors.
2. How was the 'Subjects' feature handled by the algorithm? The multi-value 'Subjects' feature varies from observation to observation, with some having four or more values in a comma-

separated list and others with fewer. It was initially thought that the 'Subjects' feature would be either split out into multiple rows or alphabetized within each observation to reduce the variety of 'Subjects' within the dataset. Ultimately, I decided differently. It was not clear from my research if the order of 'Subjects' implied a hierarchy where the first subject has higher importance and relevance than later subjects in the same observation. Alphabetizing the 'Subjects' may therefore have opened an avenue for accidental data loss.

3. How was the 'FloatingItem' feature handled by the algorithm? 'FloatingItem' designates books or media that can move across multiple locations within SPL's holdings, depending on where the item is needed by faculty or visitors. In initial testing, the algorithm was unable to properly place floating items in their last location, contributing to low accuracy scores. Floating items were then dropped because of the unpredictability of their placement in SPL's holdings. This omission from the dataset immediately improved accuracy.
4. Why was a second confusion matrix produced on 'ItemLocation' values outside of the central library? The first confusion matrix, shown in section 4.1 below, highlights a challenge with predicting 'ItemLocation' for SPL. Specifically, the Random Forest algorithm does well at placing items into the central library storage area ('cen') with a high degree of accuracy, but falls short when distinguishing between other storage areas. 'cen' represents a significant portion of the total storage space (see section 2.3 above), accounting for the bulk of the overall accuracy score. A second Random Forest was run specifically on non-'cen' data to assess accuracy. This second Random Forest was given its own confusion matrix of results.

3.3 Project Timeline

Here is the initial six-week timeline for the project:

Data exploration/pre-processing (2 weeks)	Get a better sense of the data beyond the basic descriptive statistics
Data visualization (1 week)	Visualize the results of data exploration

Random Forest (1 week)	Perform machine learning
Recording/documenting results (2 weeks)	Compile the results of the classification and visualizations into a final report

Table 5 - Initial timeline for the project.

I made only slight changes to this timeline for the project:

Week 2	Ran an initial Random Forest on 100,000 observations (out of 2.6 million) and saw results consistently lower than 46% accuracy. This first effort prompted a refocus of the project on identifying the causes of the low accuracy to attempt to boost the model.
Week 3	Started drafting the final paper to document progress as it occurred rather than at the end.
Week 4	Tried an Artificial Neural Network instead of a Random Forest, saw no improvement in overall accuracy. Re-evaluated the docs, started to add correlation testing using Python's SelectKBest for scoring accuracy.
Week 5	Dropped the 'FloatingItems that have multiple locations in the collection. Prepared draft of model code with final accuracy and confusion matrices.
Week 6	Finished full draft of the paper and the code. Shifted into paper editing.

Table 6 - Adjustments to the initial project timeline and the week in which the adjustment occured.

4. Analysis & Results

4.1 Classification outcomes: the numbers

The final testing results are as follows:

Test	**Overall accuracy**
T1 – total dataset of 2.6 million rows (2/3 1/3 split)	51.329%
T2 – non-'cen' dataset of approx. 1.1 million rows (2/3 1/3 split)	15.401%

Table 7 - Table showing tests and overall accuracy.

Once constructed, scikit's Decision Tree also calculates the Gini importance of the training set's features.

The higher the value, the greater Gini importance that the algorithm places on the feature.

	Author	**PublicationYear**	**Publisher**	**Subjects**	**ItemType**	**ItemCollection**
Gini Importance	0.1535	0.1003	0.1267	0.1550	0.070	0.3950

Table 8 - Gini Importance calculations for features in the Decision Tree, showing 'ItemCollection' to be the most important feature.

Below is the first confusion matrix for the full testing set (T1), including 'cen'.

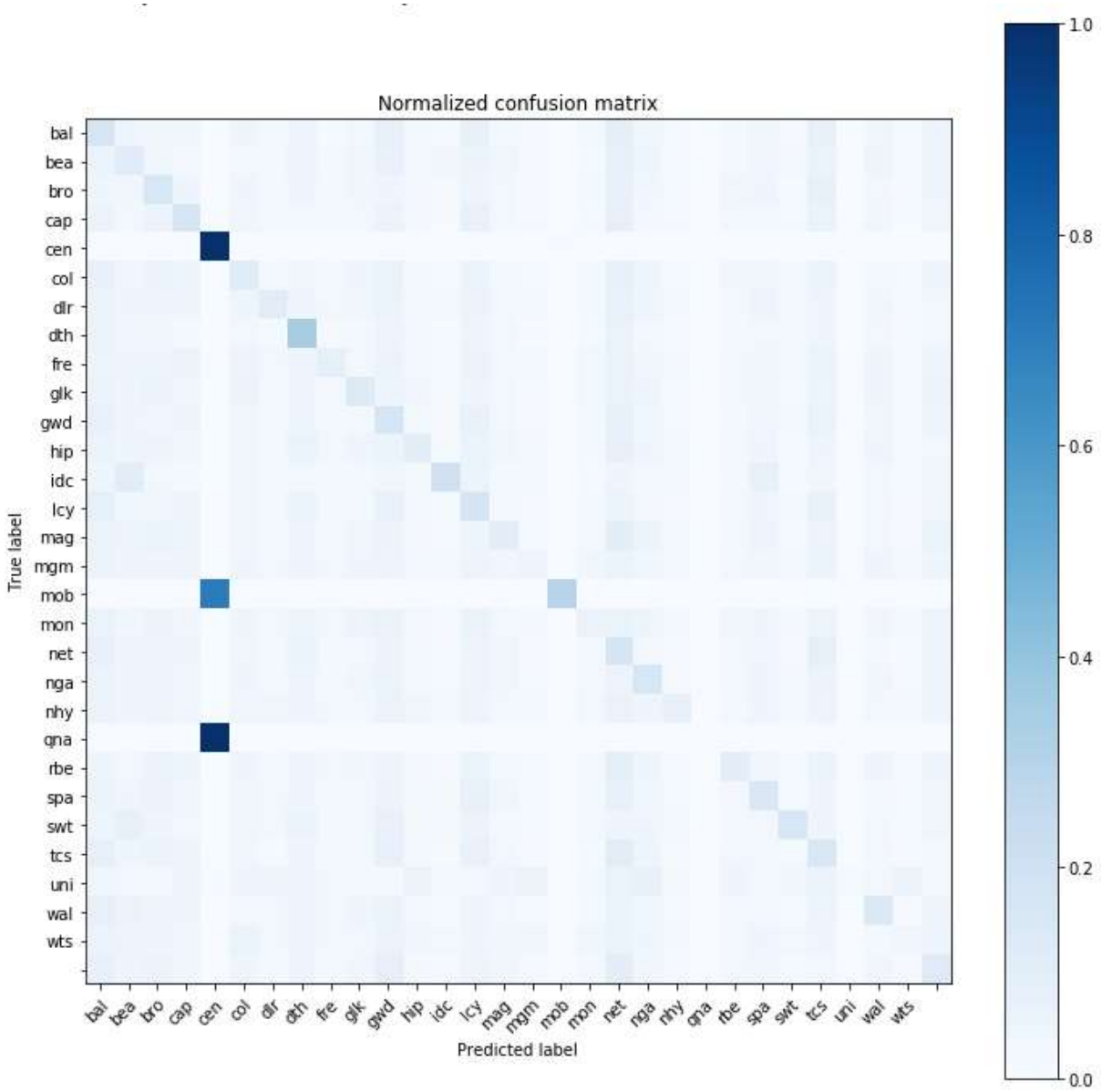

Figure 2 - Confusion matrix of the testing results for the total testing space of 2.6 million observations. Predicted and True labels correspond to the 'ItemLocation' feature and are encoded here to match the original dataset.

In Figure 3 (below), the second confusion matrix for the second testing set (T2) covers all non-‘cen’ data with a second Random Forest.

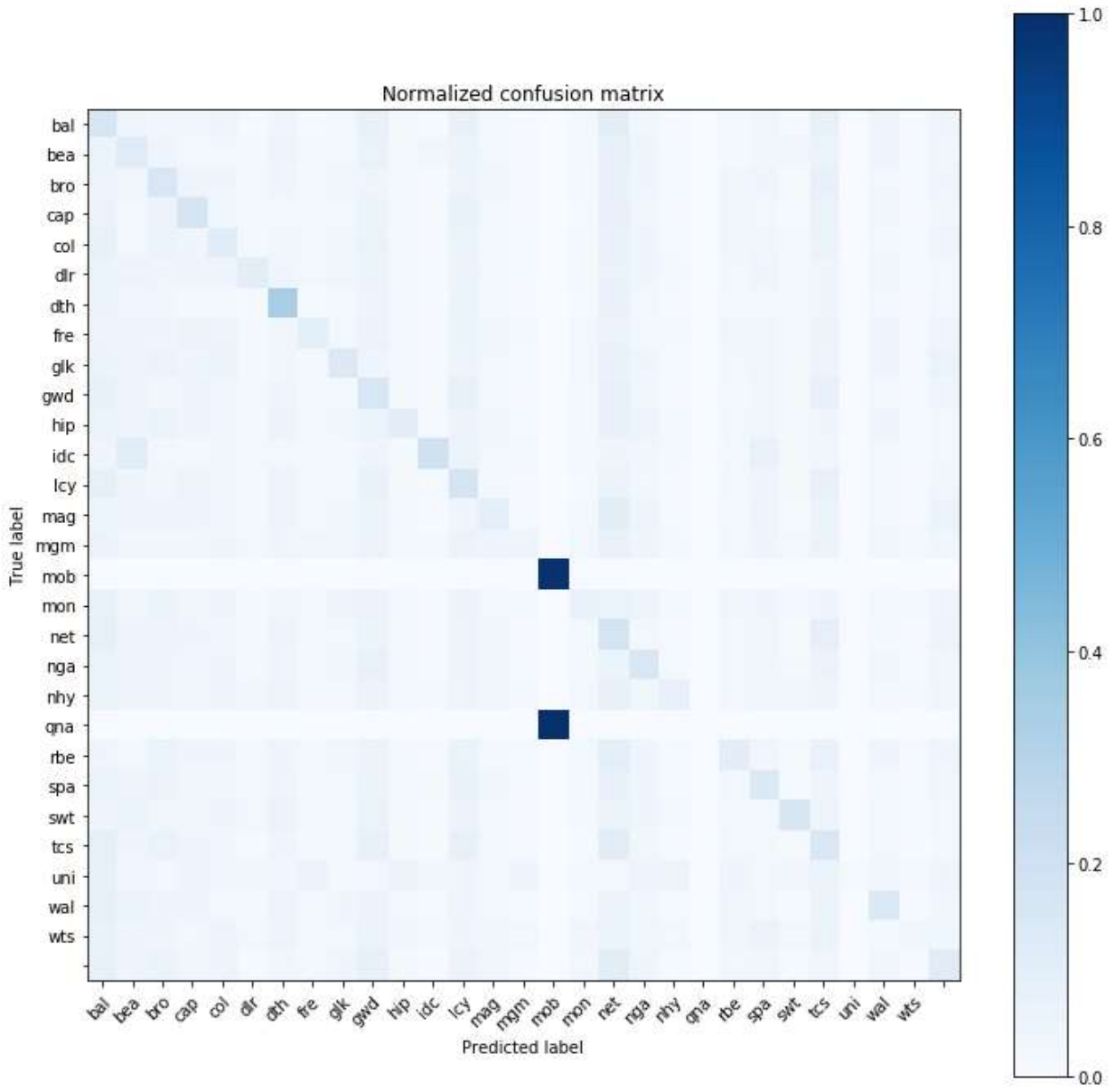

Figure 3 - Confusion matrix of the testing results for the non-'cen' testing space of approximately 1.1 million observations. Predicted and True labels correspond to the 'ItemLocation' feature and are encoded here to match the original dataset.

4.2 Classification outcomes: the analysis

The confusion matrix (Figure 2) clearly lays out the inaccuracies of the algorithm. Ideally, the diagonal from upper left to bottom right would be dark in color with most True labels matching Predicted labels. Unfortunately, that is not the case. Predicted labels are skewed to the lower end of the normalized scale, suggesting a wide spread of selections where almost every True label matches some percentage

of Predicted labels. This demonstrates the algorithm's poor performance: the algorithm is predicting a portion of every available label rather than focusing on the True label.

There are three outliers, all within the Predicted label of 'cen', the Central Library storage at 1000 4th Avenue which is the largest storage area with the greatest number of volumes by far (see Figure 1 above). The algorithm performs well when determining 'cen' vs. others in a one vs. all sense. There is another way to think about this, specifically within the context of the remaining labeling on the confusion matrix's diagonal. The algorithm performed significantly better at identifying the 'cen' than others, and since 'cen' is by far the largest storage area, it follows that the bulk of the algorithm's accuracy comes from 'cen' with very few accurate predictions for other storage areas. The second and third outliers are on the 'mob' and 'qna' true labels, often predicted by the algorithm as 'cen'. 'mob', unfortunately, is the undefined case previously described in section 2.3 above. 'qna' corresponds to Queen Anne, 400 W GARFIELD ST, and includes 47,900 observations. According to the algorithm, there is similarity in prediction between the undefined case 'mob', the Queen Anne satellite facility, and the Central Library.

Table 8's summary of the Gini Importance of each feature for the algorithm further highlights problems in accuracy. While not directly comparable, the outcomes of Gini Importance and the earlier Accuracy Score (Table 3) are in conflict. The Accuracy Score showed that 'Author' had the highest variance among features relative to the 'ItemLocation' field, suggesting that 'Author' would be the most important feature for the algorithm to consider for its predictions. This was not the case. The algorithm found that the highest Gini Importance belonged to 'ItemCollection' and then 'Subjects', placing 'Author' in third.

These insights suggest a few dynamics about the collection in relation to the algorithm. Firstly, 'cen' outweighs other storage areas by quantity, giving the algorithm more data to learn from in order

to predict the 'cen' label. Secondly, the size of 'cen' outweighing other storage areas lowers the usable data for the algorithm to differentiate other predictions, muddling results. These two insights taken together suggest a modification for this algorithm using boosting: take everything the RandomForest *would* predict as not being 'cen' and instead forward non-'cen' observations to a second algorithm specifically built from non-'cen' data. This second algorithm would thus be specifically targeted on the other smaller storage areas, potentially generating better accuracy without the weight of a single storage area shaping the data. Finally, the low accuracy of predictions of non-'cen' label suggests that, within SPL's holdings in general, the non-'cen' storage areas are insufficiently distinct to be separable by an algorithm. That is, the non-'cen' storage areas are not separated by an interpretable logic within the data.

5. Conclusion

5.1 Did it work? Complexities in cultural property management

Ultimately, the model failed to accurately predict the placement of books. Overall, the project was inconclusive. The accuracy of the total dataset testing was 51.329%. The 'cen' facility at Central Library accounts for 57.9% of the total SPL holdings. The algorithm produced in this project is therefore less accurate than an algorithm that predicts only 'cen' in all cases.

Yet there are gains to be made in understanding SPL's holdings and placement strategies based on this project. The low accuracy of the second testing on non-'cen' data (15.401%) indicates the difficulty the algorithm had at determining the differences in locations based purely on a dataset of the books and media in those locations. The non-'cen' locations are too similar to be distinguished when looking at the books and media alone. Said another way: the collections of the non-'cen' locations are so similar as to be almost indistinguishable based on the books and media on site. Taken together, the insights of the first and second dataset models highlight important logical conjectures about SPL's storage strategy.

1. There is something about the Central Library that makes it distinguishable from the other satellite facilities. It may be the size, the diversity, the subjects, or the item counts in 'cen' that distinguish it from others – but somehow, it is distinguishable. The algorithm does better than 1-in-30 random selection by a wide margin when selecting 'cen'.
2. Books do not go to unique satellite facilities based on their own characteristics. Instead, it may be that the *purpose* of the satellite facilities is to be similar in their holdings. The satellite facilities are individual libraries within the system. It follows, then, that the satellite facilities themselves should offer similar ranges of books, subjects, and media to be able to meet a wide audience at each location.

In the end, I believe the central question of this paper on the 'placement' of books within SPL's holdings was off the mark. It is not so much a question of where a book is placed but how books on similar subject, by similar authors, or from similar publishers are distributed within the satellite library system so that SPL can reach its audience equally across its total service area. This conjecture is supported by the dropped in assumed influence of the 'Author' field, which showed promise in an assessment of Accuracy Score but dropped when computing Gini Importance. There appears to be a logic at play to share the intellectual cross section of SPL's holdings widely, at least in satellite locations, and part of that sharing is to distribute Authors widely, lowering the effectiveness of that field in predicting satellite locations. SPL's push for synonymity in non-'cen' locations is itself a potential area of study – not as machine learning, but through evaluation of SPL's relationship with its community. How similar are the collections of the satellite locations relative to one another? Do they each see the same types of books checked out at the same frequency, or are there nuances depending on the neighborhoods that use those libraries? Does SPL intentionally shift certain satellite library holdings in a specific subject direction, for example skewing closer to natural sciences if a research lab or university is nearby? These questions could prompt future studies.

5.2 Where to go from here

There is more that can be done with this data to improve its accuracy.

1. Shift to a 'cen' vs. all' approach using a binary classifier like Logistic Regression. This may improve results.
2. Conduct comparative analysis between collections to see if satellite locations are, in effect, duplicates of each other with similar offerings. Related to the discussion above in section 5.1, take a closer look at the satellite locations to see what makes each unique.

That said, there is a limit to what I can do without new data.

1. Research the placement process to create the placement algorithm. This could not be extrapolated directly from the data.
2. Need data on the newest materials to look for trends in placement. This project looks at high-use items in the current SPL holdings. A dataset on purely new books is needed to more accurately assess trends in placement.
3. Find out which 'ItemLocation' options are no longer used for new material by SPL and remove them from testing. This would decrease the number of prediction options.

This project outlines a few of the challenges facing SPL placement predictions and opens a few paths to improving the algorithm in the future.

6. References

Rebecca A. Buck and Jean Allman Gilmore, eds. (2010). *Museum Registration Methods 5th Edition*. The American Association of Museums Press

City of Seattle (2019a). *Seattle Checkouts by Title: From City of Seattle Open Data*. Retrieved from https://www.kaggle.com/city-of-seattle/seattle-checkouts-by-title

City of Seattle (2019b). *Seattle Library Collection Inventory: From City of Seattle Open Data.* Retrieved from https://www.kaggle.com/city-of-seattle/seattle-library-collection-inventory

City of Seattle (2019c). *Checkouts by Title*. Retrieved from https://data.seattle.gov/Community/Checkouts-by-Title/tmmm-ytt6

City of Seattle (2019d). *Checkouts by Title (Physical Items)*. Retrieved from https://data.seattle.gov/dataset/Checkouts-by-Title-Physical-Items-/3h5r-qv5w

City of Seattle (2019e). *Integrated Library System (ILS) Data Dictionary.* Retrieved from https://data.seattle.gov/Community/Integrated-Library-System-ILS-Data-Dictionary/pbt3-ytbc

City of Seattle (2019f). *Library Collection Inventory*. Retrieved from https://data.seattle.gov/Community/Library-Collection-Inventory/6vkj-f5xf

City of Seattle (2019g). *Seattle Integrated Library System (ILS) Dictionary: From City of Seattle Open Data*. Retrieved from https://www.kaggle.com/city-of-seattle/seattle-integrated-library-system-ils-dictionary

The Minitab Blog (2016). *Understanding Analysis of Variance (ANOVA) and the F-test*. Retrieved from https://blog.minitab.com/blog/adventures-in-statistics-2/understanding-analysis-of-variance-anova-and-the-f-test

Dan Murphy (2019). *Review of Seattle Library Checkout Records*. Retrieved from https://www.kaggle.com/murphydan/review-of-seattle-library-checkout-records

Dan Ofer (2019). *Library Checkouts EDA*. Retrieved from https://www.kaggle.com/danofer/library-checkouts-eda

Seattle Public Library (2017). *Seattle Library Checkout Records: Twelve years of checkout records*. Retrieved from https://www.kaggle.com/seattle-public-library/seattle-library-checkout-records

Marcellus Turner (n.d.). *Welcome: Welcome to The Seattle Public Library*. Retrieved from https://www.spl.org/about-us/the-organization/leadership/welcome

Scikit-learn (2019). *Model evaluation: Accuracy score.* Retrieved from https://scikit-learn.org/stable/modules/model_evaluation.html#accuracy-score

7. Appendix A: Full Feature List

BibNum	
Type: Numeric, categorical, single-value	
Description: *The unique identifier for a cataloged item within the Library's Integrated System (ILS).*	
Count: 2687149	Unique: 584391
Most frequent value: 1923072 (128 count)	
Note: There are only 584,391 BibNum values within the 2.69 million observations of the collection. This is due to repetition of BibNums for multiple copies of the same item. For example, BibNum 1923072 corresponds to “Guinness World Records” (c200 edition) which is included 128 times in the collection.	

--

Title	
Type: Nominal, single-value	
Description: *The full title of an item.*	
Count: 2672825	Unique: 567632
Most frequent value: Uncataloged Adult book (214 count)	
Note: The count of Titles is lower than the count of BibNums, suggesting that there are some items with BibNums but no Title. There are also Titles provided by Library staff or data managers like “Uncataloged Adult book,” which occurs 214 times. All told, there are 567633 unique Titles, which is close but not equivalent to the number of unique BibNums (584391)	

--

Author	
Type: Nominal, single-value	
Description: *The name of the first author of the title, if applicable.*	
Count: 2259761	Unique: 217936
Most frequent value: Patterson, James, 1947- (4580 count)	
Note: The count of items with Authors is lower than the total items, suggesting a portion of the collection that lacks confirmed authorship. The lower count of Unique values is evidenced by the number of books by common authors: authors like James Patterson can appear more than one thousand times in the collection.	

--

ISBN	
Type: Numeric, categorical, multi-value	
Description: *Comma-delimited list of ISBN(s) for this title.*	
Count: 2098618	Unique: 397036
Most frequent value: 1631212583, 9781631212581 (364 count)	
Note: The International Standard Book Number, assigned by one of a number of different International Organization for Standardization (usually abbreviated ISO) partners, that distinguishes the unique identify of a single published book. The unique count of ISBNs is lower than that of the	

unique count for BibNum, which is logical on the surface – more books have been printed than there are ISBNs to match – but misleading in the reality of the data. As evidenced by the most frequent value of this feature, ISBNs for different editions of the same book may be listed together within one BibNum. This muddles the accuracy of the unique count.

--

PublicationYear	
Type: Nominal, multi-value	
Description: *Date(year) of publication.*	
Count: 2655923	Unique: 15289
Most frequent value: [2016] (151025 count)	
Note: Format is key. Here are some examples copied from the SPL FAQs page: - 2005 = publication date - c. 2005 = copyright symbol - [2005] = printing date - p. 2005 = phonogram copyright symbol - 2004, c. 2005 = publication and copyright date - 2005-2007 = intervening years - [2005?] = approximate date At first glance, the data here appear to be inconsistently formatted. But as the examples show, the formatting is essential for communicating details about the data.	

--

Publisher	
Type: Nominal, multi-value	
Description: *The name of the publishing company for this item.*	
Count: 2649615	Unique: 96419
Most frequent value: Random House, (35353 count)	
Note: For some items, the Publisher has been guessed without being formally identified. Ex: [Dept. of External Affairs?]. Without seeing the database system underlying this data, it is difficult to confirm if publisher names are controlled by keys or added ad hoc. Are there misspellings of Random House in the collection? Are those then decreasing the accuracy of the data overall?	

--

Subjects	
Type: Nominal, multi-value	
Description: *A comma-separated list of the subject authority records associated with the title, including Motion Pictures, Computer Programming, etc. Typically these are highly specific.*	
Count: 2621275	Unique: 439811
Most frequent value: Rock music 2011 2020 (10945 count)	
Note: The total count of this feature is misleading. The fourth most frequent value is “Popular music 2011 2020, Rock music 2011 2020” with 5033 occurrences. This demonstrates that subjects are variably listed, from zero or one to several or more in a single item. There are items where “Subject A, Subject B” are not alphabetized, leading to double-counting of “Subject B, Subject A”. However,	

alphabetizing will change the order of subjects which may, in some collections, represent a hierarchy with primary subjects coming before others. For this reason, Subjects were not alphabetized or separated for testing.

--

ItemType	
Type: Nominal, encoded, single-value	
Description: *Horizon item type. Look up value descriptions in HorizonCodes using CodeType "ItemType".*	
Count: 2687149	Unique: 63
Most frequent value: Book: Adult/YA (1338207 count)	
Note: This is an encoded feature. In the source dataset, ItemType is listed in codes like “acbk” and “jcbk,” presumably to save on the storage footprint of the total collection. These are somewhat interpretable from first sight (“acbk” ➔ adult collection of books ➔ “Book: Adult/YA”), however the data dictionary makes substitution straightforward for use in the dataset. Each observation belongs to only one ItemType, with “Book: Adult/YA” greatly outnumbering the rest.	

--

ItemCollection	
Type: Nominal, encoded, single-value	
Description: *Collection code for this item. Look up value descriptions in: Integrated Library System (ILS) Data Dictionary.*	
Count: 2687149	Unique: 193
Most frequent value: CA-Reference (384170 count)	
Note: This is an encoded feature. In the source dataset, ItemType is listed in codes like “canf” and “nanf” presumably to save on the storage footprint of the total collection. These are somewhat interpretable from first sight (“canf” ➔ “CA Reference”), however the data dictionary makes substitution straightforward for use in the dataset. Each observation belongs to only one ItemCollection.	

--

FloatingItem	
Type: Boolean, single-value	
Description: *Label that indicates if an item floats.*	
Count: 2687149	Unique: 2
Most frequent value: Floating (411108 count) (other 2276041 are nulls)	
Note: An item that “floats” is moved periodically from one storage location to another, according to the data dictionary. These items make the librarian’s lives potentially easier as books increasing in popularity can be kept closer to the main facility, however it presents a significant challenge for a machine learning algorithm trying to classify a specific location. For this project, FloatingItems are not included in the analysis.	

--

ItemLocation	
Type: Nominal, encoded, single-value	
Description: *Location that owned the item at the time of snapshot. 3-letter code. Note: as of 2017, some items are "FLOATING" which means they don't necessarily belong to a specific branch. Location of given copy could change based on where the item is returned.*	
Count: 2687149	Unique: 33
Most frequent value: Central Library, 1000 4TH AV (1080480 count)	
Note: This is an encoded feature. In the source dataset, ItemType is listed in codes like "cen" and "net" presumably to save on the storage footprint of the total collection. These are somewhat interpretable from first sight ("cen" ➔ "Central Library"), however the data dictionary makes substitution straightforward for use in the dataset. Each observation belongs to only one ItemLocation.	

--

ReportDate	
Type: Date, single-value	
Description: *The date when this item count was collected from the ILS (Horizon).*	
Count: 2687149	Unique: 2
Most frequent value: 09/01/2017 (1343674 count) (10/01/2017 is 1343475 count)	
Note: There are only two dates present here, both listed above as frequent. These dates reflect when the data were digitally pulled from SPL's database.	

--

ItemCount	
Type: Numeric, single-value	
Description: *The number of items in this location, collection, item type, and item status as of the report date.*	
Count: 2687149	Unique: 169
Most frequent value: 1 (2370856 count)	
Note: By far, most items in the SPL collection have an ItemCount of 1. This is misleading. As shown in the BibNum summary where "Guinness World Records" c2000 had 128 separate observations, there is a lack of clarity between items with multiple BibNums and items with one BibNum and multiples of ItemCount. This gray area of the collection may not impact a total count of items (seven separately listed 1s is equivalent to one individually listed 7), but it adds structural uncertainty to the dataset itself – especially when considered alongside the overlapping ISBNs of some publications. The cataloging over the history of the collection does not appear to be consistent.	

8. Appendix B: Code Package

The total package for this project includes this document as well as the following files:

- *Zange_634Final_Code.ipynb* - iPython file containing final version of code
- */.ipynb_checkpoints* - directory of supporting files for the iPython file
- *Zange_634Final_README.txt* - README file for coding elements

Data files have not been included due to the large file sizes of library data.

The code cannot be run without encountering an error, since the code requires library data to run. That said, the output of the code can be viewed by opening the iPython file within a coding environment like Jupyter. Please open the README file before opening the code.