Capstone 2 Final Report

Code: https://github.com/samwrite/Springboard/blob/master/Capstone%202/Capstone%202%20Final%20Code.ipynb

By Samuel Ma

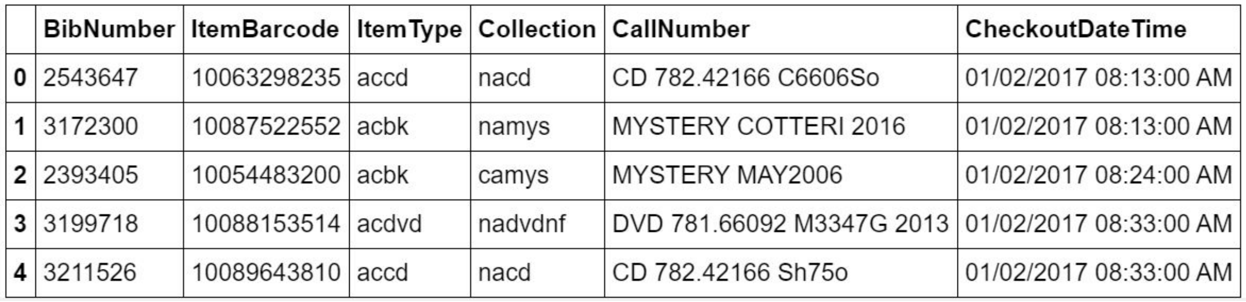
Problem Statement: What are the most popular items checked out at the library?

Link to data: https://www.kaggle.com/seattle-public-library/seattle-library-checkout-records

In this report, I will be taking data from the Seattle public library system and see if I can find any trends in regards to popularity. Although there is data from 2005 to 2017, I will only take data from one year, 2017, due to the size of the file and the lower processing power of my resources.

Data Cleanup:

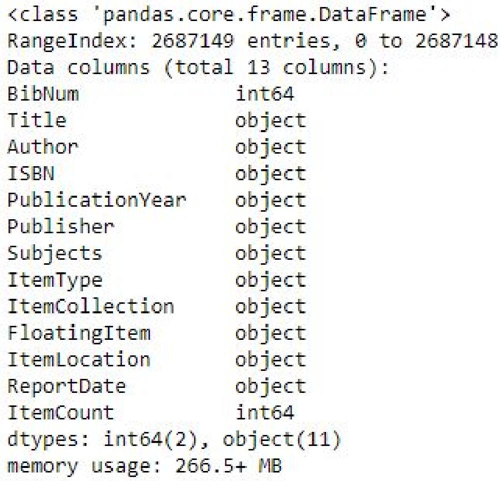
Our goal is to combine the checkout data from 2017 and the Library inventory data into one clean dataframe. Looking at the checkout data first:



	BibNumber	ItemBarcode	ItemType	Collection	CallNumber	CheckoutDateTime
0	2543647	10063298235	accd	nacd	CD 782.42166 C6606So	01/02/2017 08:13:00 AM
1	3172300	10087522552	acbk	namys	MYSTERY COTTERI 2016	01/02/2017 08:13:00 AM
2	2393405	10054483200	acbk	camys	MYSTERY MAY2006	01/02/2017 08:24:00 AM
3	3199718	10088153514	acdvd	nadvdnf	DVD 781.66092 M3347G 2013	01/02/2017 08:33:00 AM
4	3211526	10089643810	accd	nacd	CD 782.42166 Sh75o	01/02/2017 08:33:00 AM

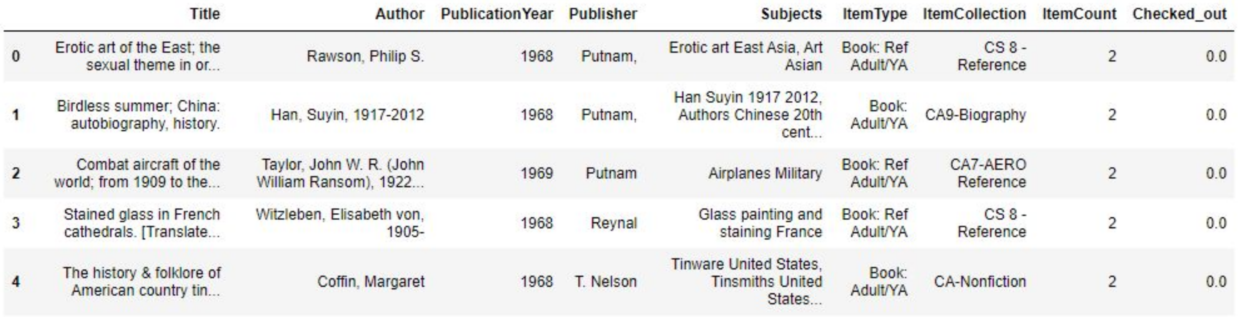
Since ItemBarcode and CallNumber are simply references/identifiers we can remove those columns.

Now we look at the inventory data:



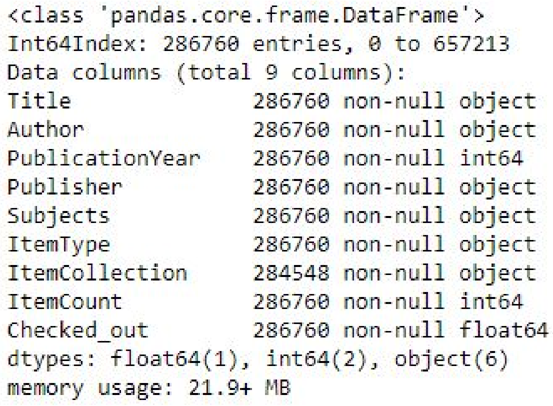
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2687149 entries, 0 to 2687148
Data columns (total 13 columns):
BibNum             int64
Title              object
Author             object
ISBN               object
PublicationYear    object
Publisher          object
Subjects           object
ItemType           object
ItemCollection     object
FloatingItem       object
ItemLocation       object
ReportDate         object
ItemCount          int64
dtypes: int64(2), object(11)
memory usage: 266.5+ MB
```

Since we will end up merging this data with the checkout data, there are a lot of duplicate columns we can remove. We will also combine all counts of an item regardless of location and report date to get the most updated ItemCount.



	Title	Author	PublicationYear	Publisher	Subjects	ItemType	ItemCollection	ItemCount	Checked_out
0	Erotic art of the East; the sexual theme in or...	Rawson, Philip S.	1968	Putnam,	Erotic art East Asia, Art Asian	Book: Ref Adult/YA	CS 8 - Reference	2	0.0
1	Birdless summer; China: autobiography, history.	Han, Suyin, 1917-2012	1968	Putnam,	Han Suyin 1917 2012, Authors Chinese 20th cent...	Book: Adult/YA	CA9-Biography	2	0.0
2	Combat aircraft of the world; from 1909 to the...	Taylor, John W. R. (John William Ransom), 1922...	1969	Putnam	Airplanes Military	Book: Ref Adult/YA	CA7-AERO Reference	2	0.0
3	Stained glass in French cathedrals. [Translate...	Witzleben, Elisabeth von, 1905-	1968	Reynal	Glass painting and staining France	Book: Ref Adult/YA	CS 8 - Reference	2	0.0
4	The history & folklore of American country tin...	Coffin, Margaret	1968	T. Nelson	Tinware United States, Tinsmiths United States...	Book: Adult/YA	CA-Nonfiction	2	0.0

Now we merge both dataframes with the BibNum column as the merging index, and then dropping that BibNum since it will no longer serve any purpose for us. In the end we will reset the index of the dataframe to make everything look nice. We also added a Checked_out column to show which items from the overall inventory were checked out.

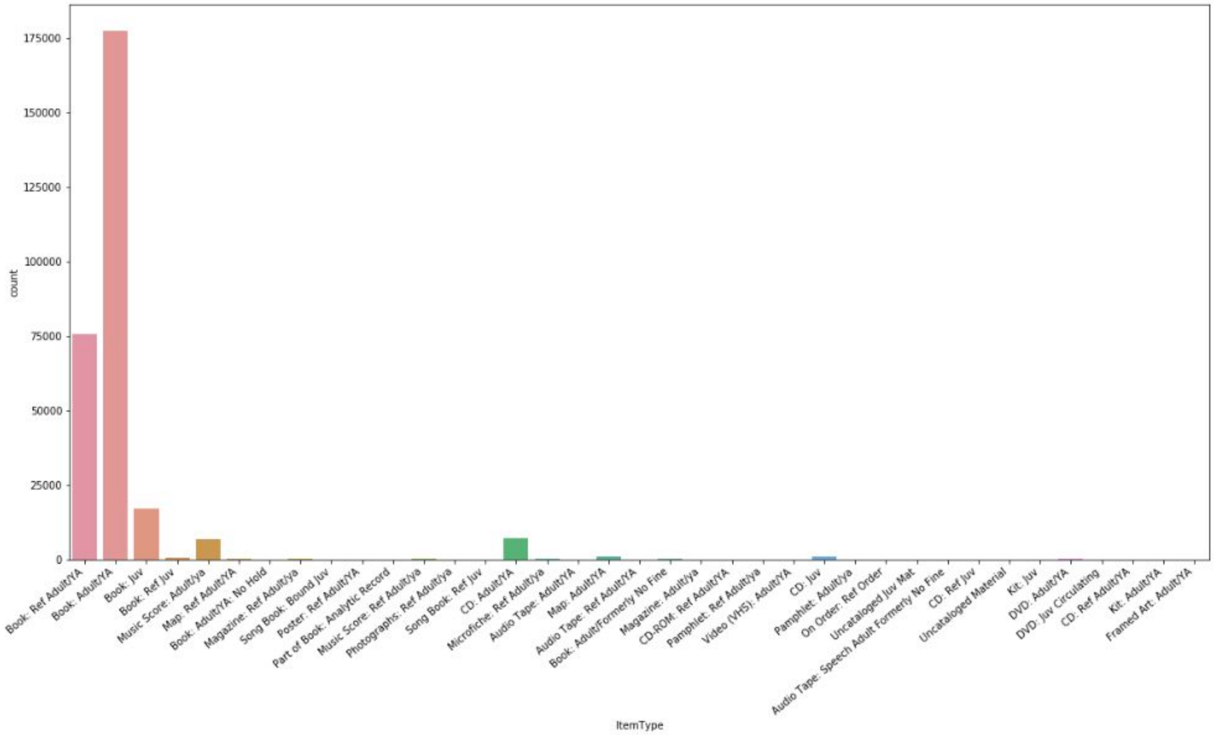


```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 286760 entries, 0 to 657213
Data columns (total 9 columns):
Title              286760 non-null object
Author             286760 non-null object
PublicationYear    286760 non-null int64
Publisher          286760 non-null object
Subjects           286760 non-null object
ItemType           286760 non-null object
ItemCollection     284548 non-null object
ItemCount          286760 non-null int64
Checked_out        286760 non-null float64
dtypes: float64(1), int64(2), object(6)
memory usage: 21.9+ MB
```

With ItemType and Collection we use the Item Data dictionary to map each code to a longer description of what the item actually is.

EDA

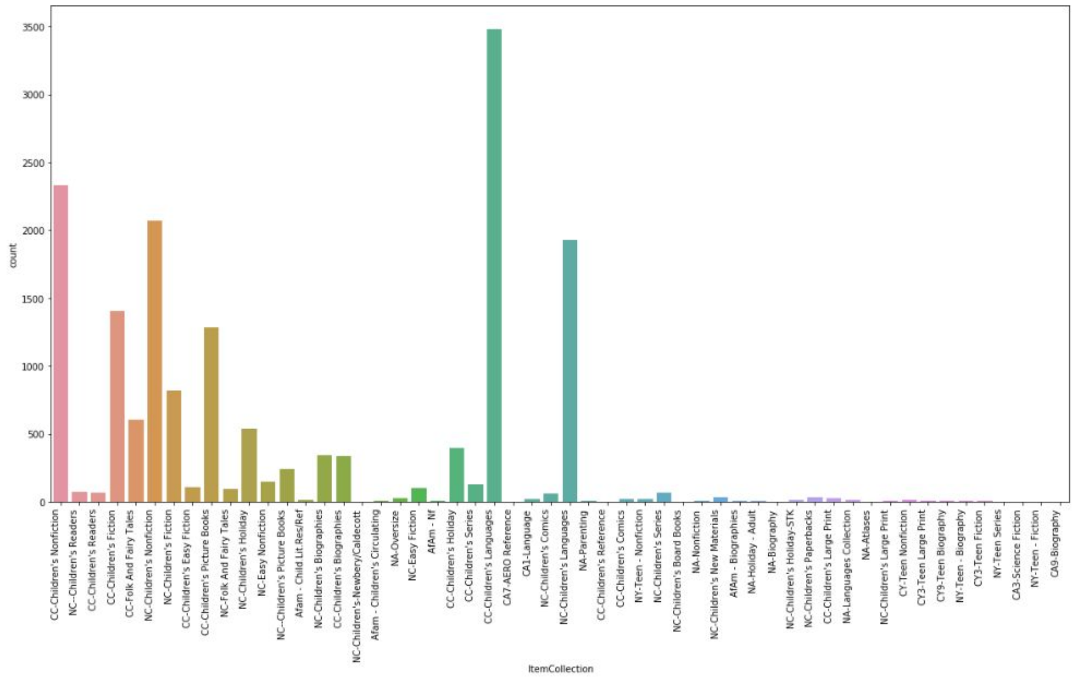
With all the data merged and cleaned up, we can take a look at which items were the most popular.



The first two columns are clearly frontrunners in most checked out items. To add more variety we will add the “Book: Juv” category into this study as well.

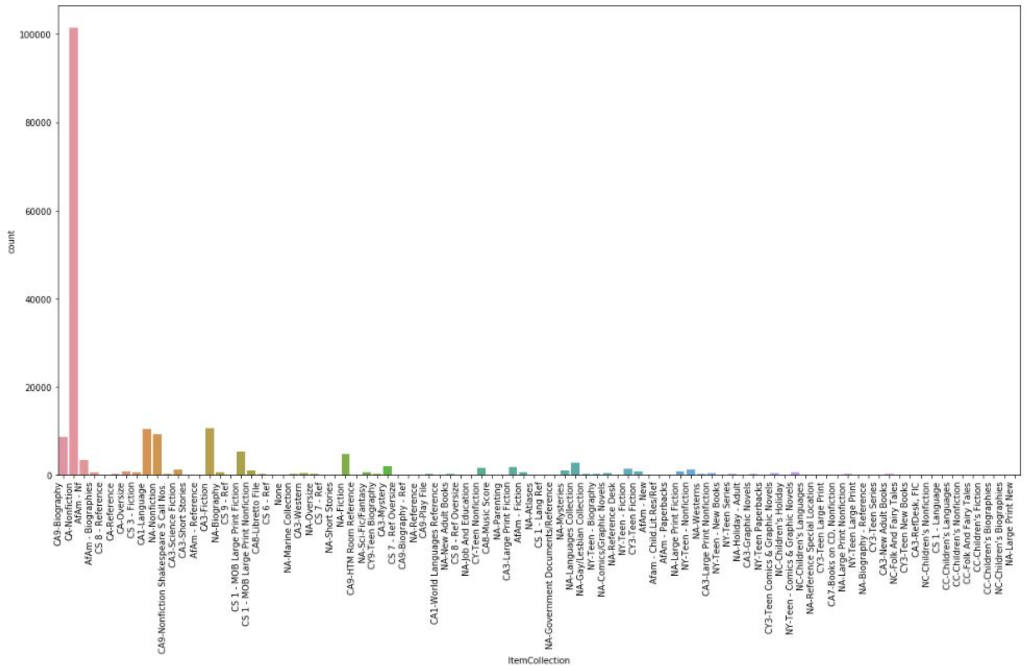
Collection Type Counts

Juvenile Book Data:



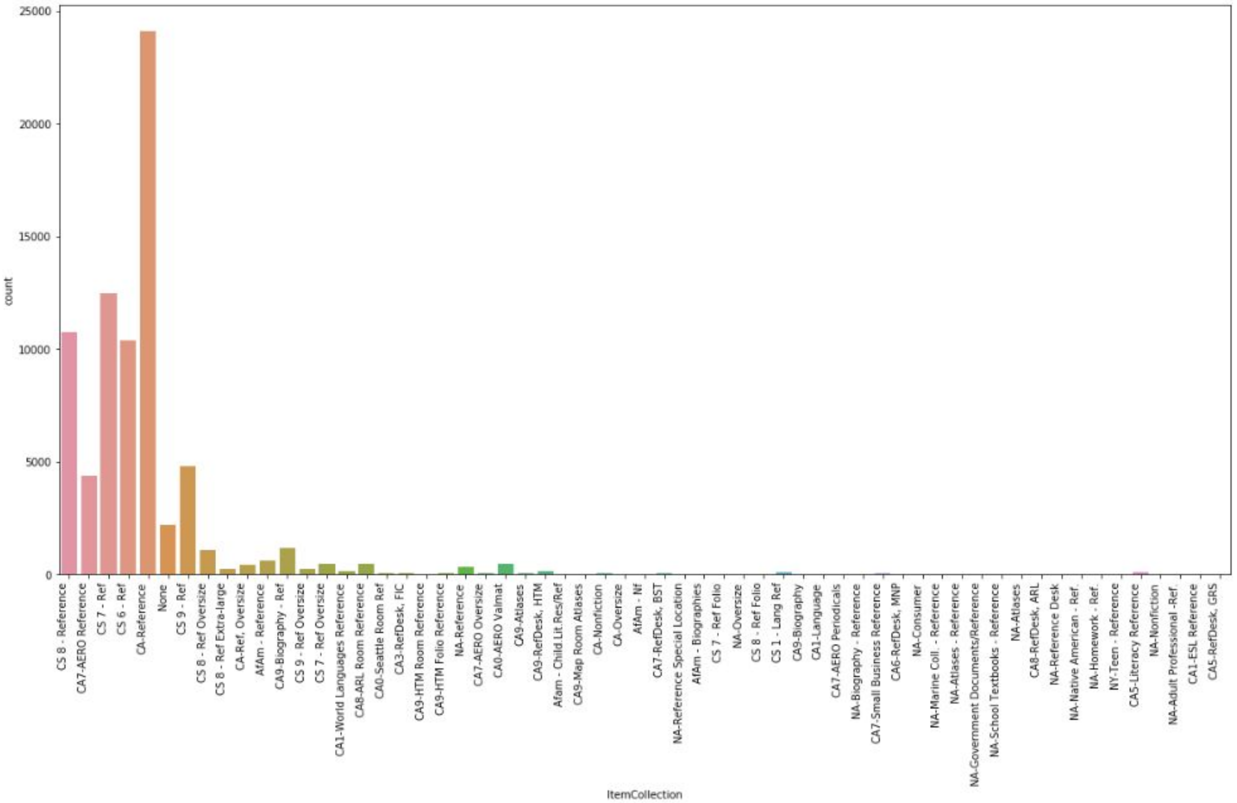
Children’s Language books seem to be the most popular in this category, with Nonfiction books coming up pretty close.

Adult Book Data:



Nonfiction books are the dominant genre for this category, around 10x more than any other single collection type.

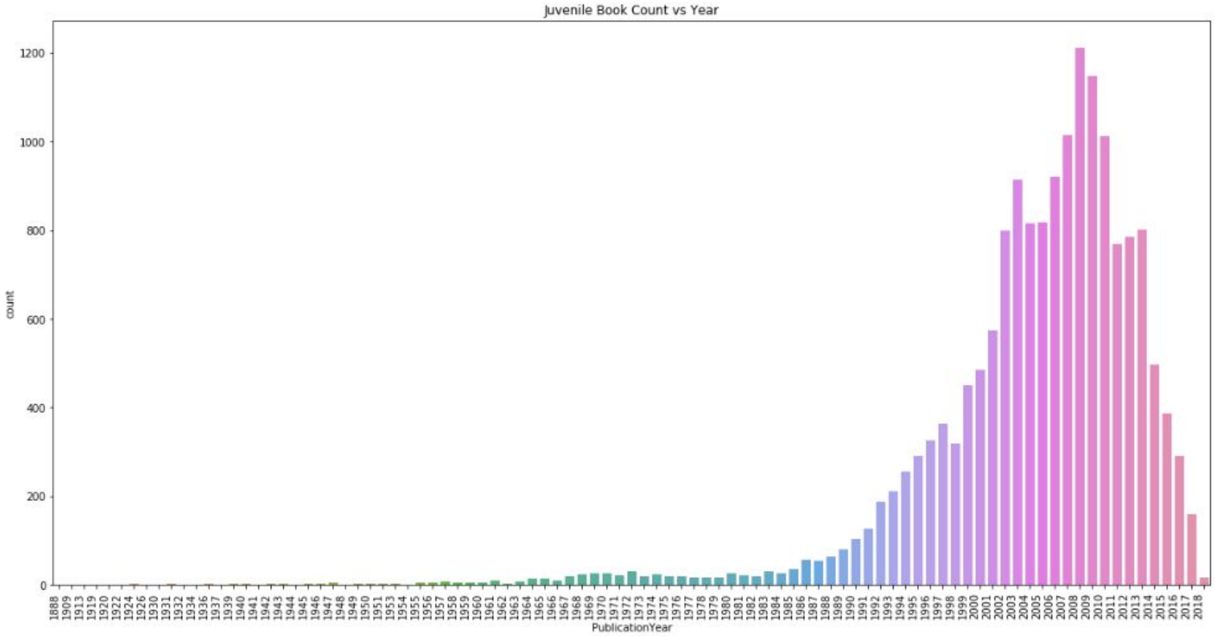
Reference Data:



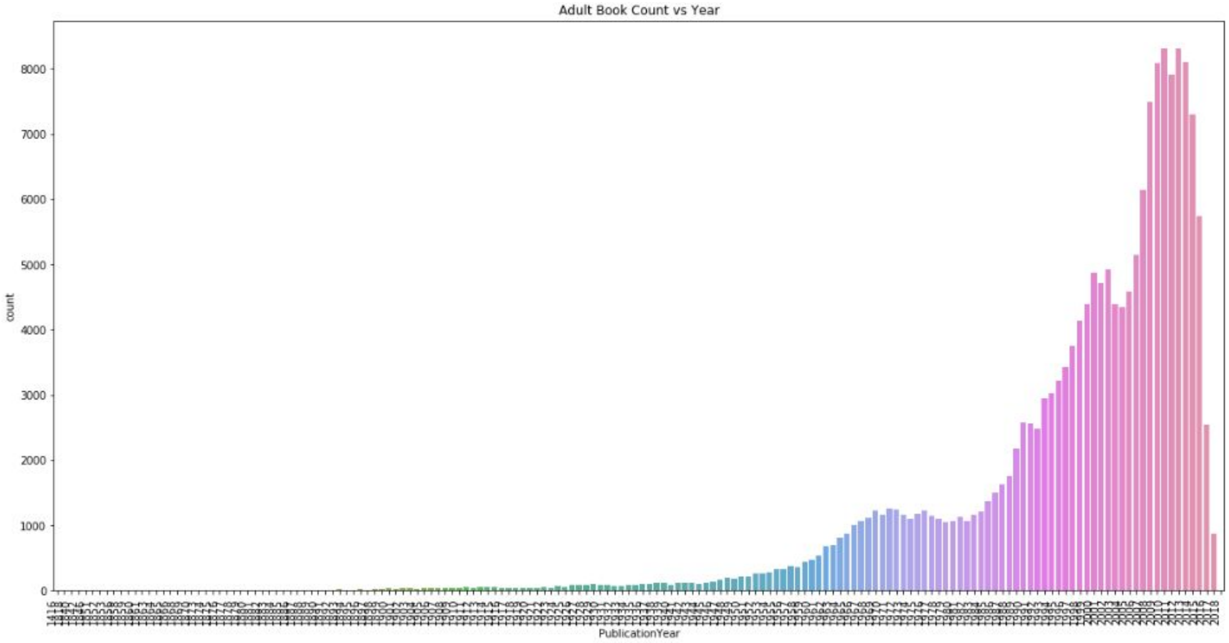
This one is a little harder to pick out exactly what was the most popular in the category, due to the Item Collection names.

Publication Year Counts

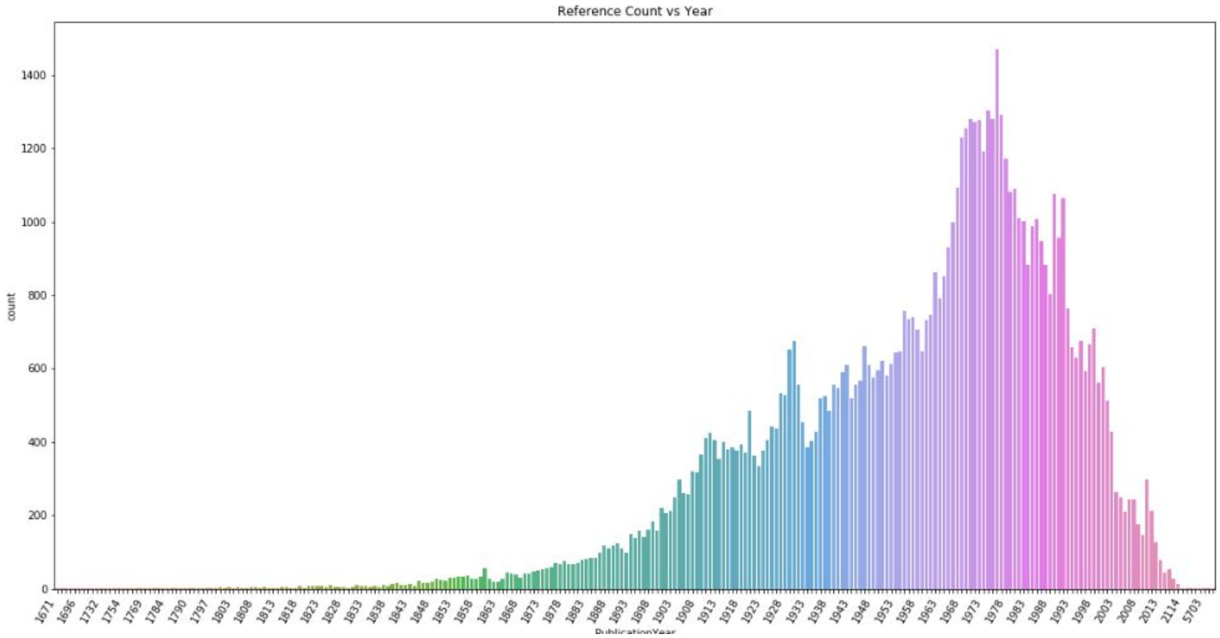
Juvenile Book Data:



Adult Book Data:



Reference Data:

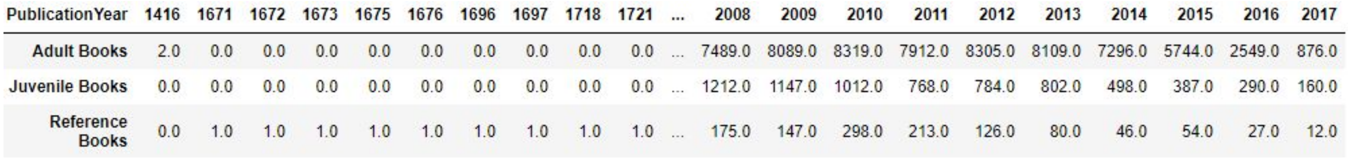


Overall there is generally an increase in books being published over the years. This makes sense due to more authors and organizations supplying more stories/reference books for a higher demand.

Stats

Chi-square test: compare year and type of item

1. Hypothesis Null Hypothesis (H0): Variables are independent. Alternative Hypothesis (H1): Variables are not independent
2. Use alpha value of 0.05



PublicationYear	1416	1671	1672	1673	1675	1676	1696	1697	1718	1721	...	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
Adult Books	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	7489.0	8089.0	8319.0	7912.0	8305.0	8109.0	7296.0	5744.0	2549.0	876.0
Juvenile Books	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	1212.0	1147.0	1012.0	768.0	784.0	802.0	498.0	387.0	290.0	160.0
Reference Books	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	...	175.0	147.0	298.0	213.0	126.0	80.0	46.0	54.0	27.0	12.0

chi-squared value: 119541.4409103959

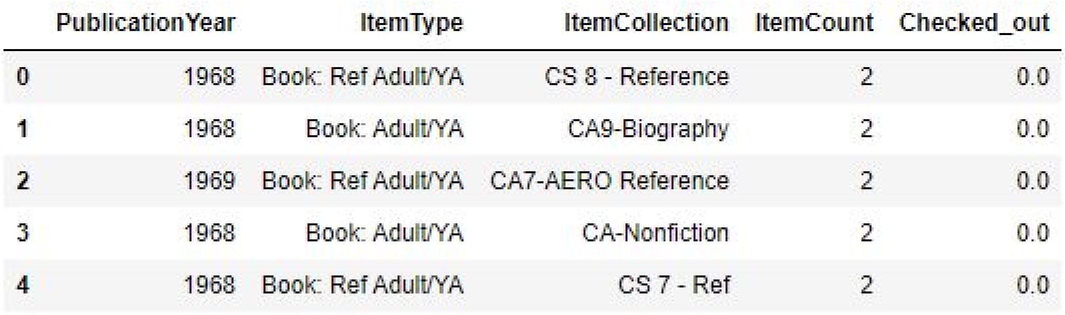
p-value: 0

degree of freedom: 510

High chi-squared value is most likely due to our larger set of data. Our p-value being 0 (most likely not exactly 0 but a ridiculously low number) further shows that the item type and publication year are associated.

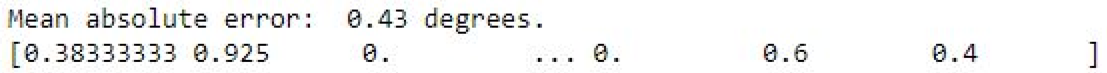
Machine Learning Models

Feature Importance - Random Forests

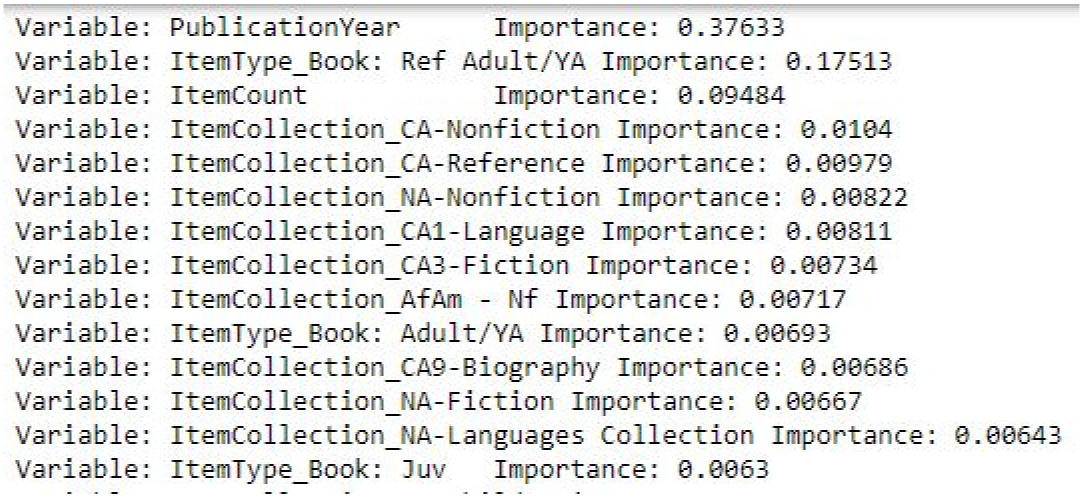


	PublicationYear	ItemType	ItemCollection	ItemCount	Checked_out
0	1968	Book: Ref Adult/YA	CS 8 - Reference	2	0.0
1	1968	Book: Adult/YA	CA9-Biography	2	0.0
2	1969	Book: Ref Adult/YA	CA7-AERO Reference	2	0.0
3	1968	Book: Adult/YA	CA-Nonfiction	2	0.0
4	1968	Book: Ref Adult/YA	CS 7 - Ref	2	0.0

We use dummy variables for our categorical variables (ItemType and Item Collection) to enable us to run random forest regression and for future classification algorithms.



```
Mean absolute error:  0.43 degrees.
[0.38333333 0.925      0.         ... 0.         0.6        0.4       ]
```



```
Variable: PublicationYear      Importance: 0.37633
Variable: ItemType_Book: Ref Adult/YA Importance: 0.17513
Variable: ItemCount            Importance: 0.09484
Variable: ItemCollection_CA-Nonfiction Importance: 0.0104
Variable: ItemCollection_CA-Reference Importance: 0.00979
Variable: ItemCollection_NA-Nonfiction Importance: 0.00822
Variable: ItemCollection_CA1-Language Importance: 0.00811
Variable: ItemCollection_CA3-Fiction Importance: 0.00734
Variable: ItemCollection_AfAm - Nf Importance: 0.00717
Variable: ItemType_Book: Adult/YA Importance: 0.00693
Variable: ItemCollection_CA9-Biography Importance: 0.00686
Variable: ItemCollection_NA-Fiction Importance: 0.00667
Variable: ItemCollection_NA-Languages Collection Importance: 0.00643
Variable: ItemType_Book: Juv   Importance: 0.0063
```

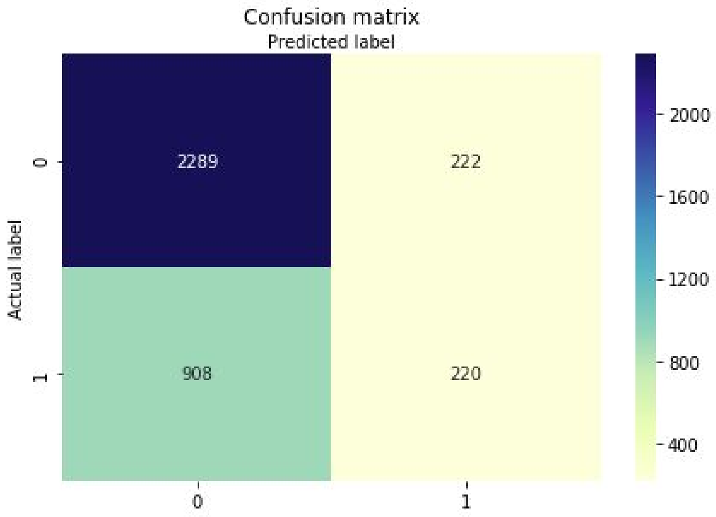
Here we see that the highest factor in determining checkouts was Publication Year at nearly 38%. We include Item Type for the three categories we analyzed above to see how much of a factor they were. Item Count held a 10% factor in determining checkouts.

Decision Trees

We used the same X and y training/testing data and list of features to see how well Decision Trees would predict our data. We ended up with an accuracy of Accuracy: 0.5221214619400935, which rounded up is 53%, not too bad but it is not the best as well. You can refer to the jupyter notebook code for the implementation.

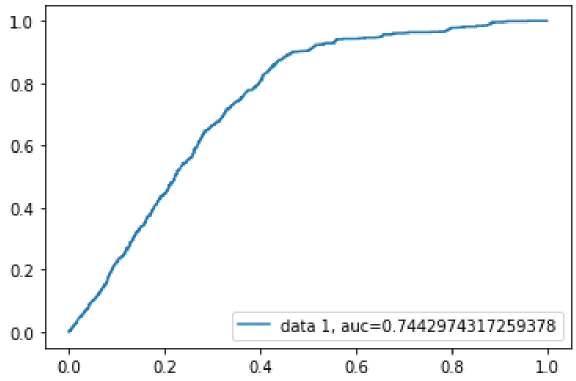
Logistic Regression

Again, reusing X and y training/testing data we can create a confusion matrix after applying a Logistic Regression model.



Accuracy: 0.6894751305303655
Precision: 0.497737556561086
Recall: 0.1950354609929078

We end up with an accuracy of 69% which is better than our Decision Tree model.
Below is an ROC curve of the data.



Conclusions and Moving Forward

Based on the results from our Random Forests, we see that Publication Year plays the biggest part in determining library checkouts. With the rise of shared digital libraries and online resources in recent years, it would be interesting to see less published material in libraries in subsequent years.

Things we can consider to look at if we were to move forward with this project would be looking at the locations of where the checkouts happen as well as looking more in depth at inventory count. We can use these two factors to help determine whether or not resources should be allocated in order for highest geographic efficiency. It may also be worthwhile to look at specific checkout times and find any trends that may happen during different seasons, for example, would more children's books be checked out during the summer due to summer reading programs.