Youdotcom Wikimedia Analysis doc

DP

Context

Analysis focuses on analyzing a Wikimedia clicks dataset that tells you where users are navigating from and to, and the number of people in a given month that are going from one link to another.The technical implementation of the project is that we:

Exploratory questions

1. **[Scope]** the exploration for December 2023's english dataset for wikipedia clickstream data
2. **[Exploratory Data Analysis]** We have wiki clickstream data for articles that users are navigating to and from, we want to understand the basic facts from Exploratory Data Analysis
3. **[Article exploration]** Interesting to profile the top 10 links that users are navigating from, to and the transitions users are making
4. **[Category exploration]** Then, we try to map these links to categories and find the top category transitions for Wikipedia categories
5. **[Predict Category]** We now Predict the next category from previous category with reasonable accuracy

Findings

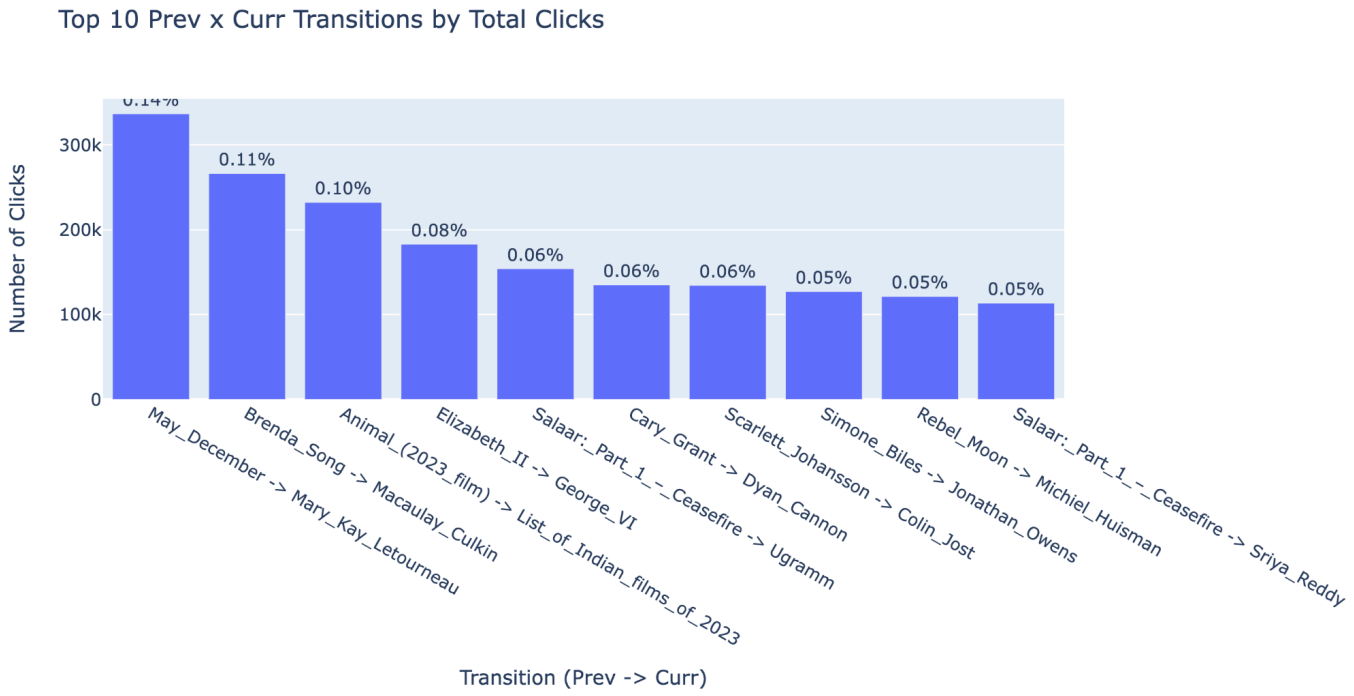
Users landing Film Wikipedia Articles are most likely to come from external Search, our largest inbound channel.

1. **Traffic composition** of December 2023 data
    a. 66% of the links are external, 33% internal
    b. Source of redirects to Wiki articles
        i. Previous articles redirecting to Wikipedia articles
            1. 48% of the previous article links are from external search sites like Google, Bing, etc.
            2. **Top Previous** Among the ones that were internal, Wiki Home page takes ~1% of the traffic where people redirect from, and films (Animal, Salaar, Wonka) occur as a theme among the top articles, indicating a trend that users may land on Wikipedia to find a Film's information.
        ii. Top 10 Wiki Articles landed upon include authors, football leagues, celebrity names and even Porn sites from the random sample

Youdotcom Wikimedia Analysis doc

DP

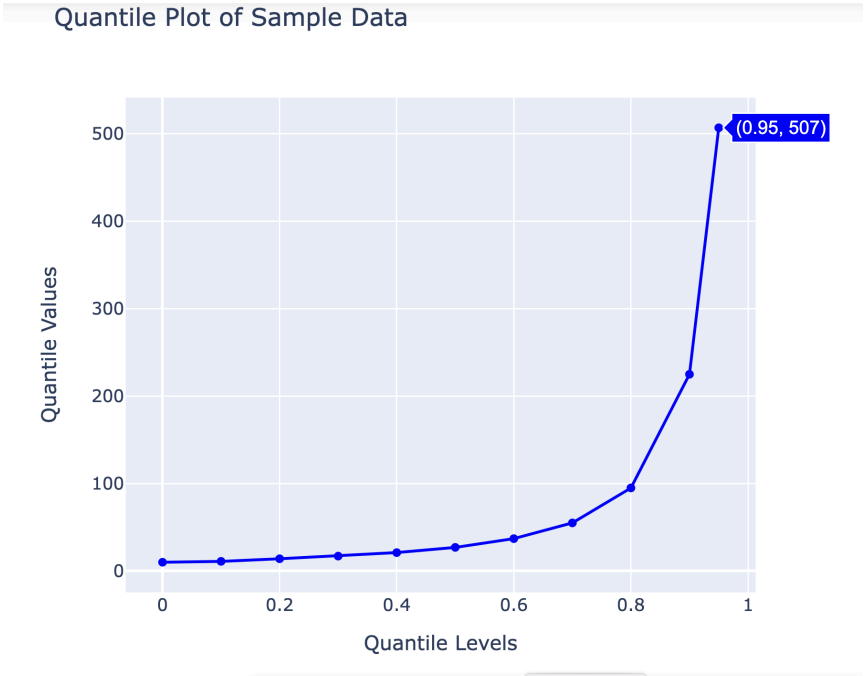
Exhibit showing top 10 transitions (prev -> current article)



2. **Quantile analysis**
    a. **95th percentile** of transitions have >= 500 counts in December 2023
    b. **Composition of >95% data**
        i. In overall data, we see 2% composition linking from External -> Main Page of Wikipedia
        ii. In the sample data we see that 33% composition is linking from External -> List of highest Grossing Indian films
    c. **This quantile plot generalizes in sample data and overall data** and we had higher confidence of using our 10% random sample for profiling overall trends

Youdotcom Wikimedia Analysis doc

DP



3. **Categorizing articles and finding trends**
    a. Mined articles for top 500 tokens among previous and current articles, and manually mapped predefined categories such as Films that may contain different popularly occurring tokens

'Film': ['film', 'actor', 'actress', 'director', 'cinema', 'tv_series', 'movie', 'disney', 'marvel_comics',

'soundtrack', 'franchise', 'season', 'series', 'spider', 'star_wars', 'screenplay'],

    b. Categories from where traffic originates from and lands on, respectively
        i. **Source** Almost 95% is not mapped to Wikipedia article category 48% is Search, 47% external (other sites), 2% start from Wiki Film articles
        ii. **Landing article** 44% is Film, 32% redirects back to Search and 11% redirects to a music article.
        iii. **Search / External is a prominent originating category and Film Articles on Wikipedia are a prevalent landing category**
        iv. Search (Google, Bing) -> Film transitions account for 35% of all transitions

Exhibit showing donut charts for (i) Originating article on left and (ii) landing article on the right

Originating Article **Landing Article**

Youdotcom Wikimedia Analysis doc

DP

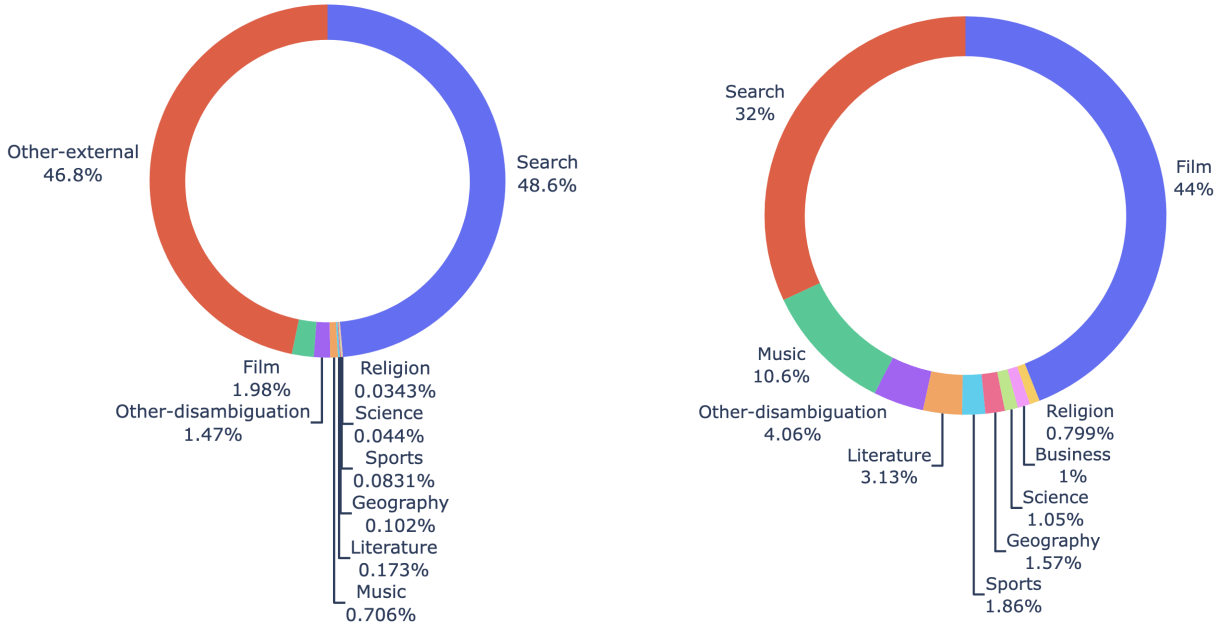
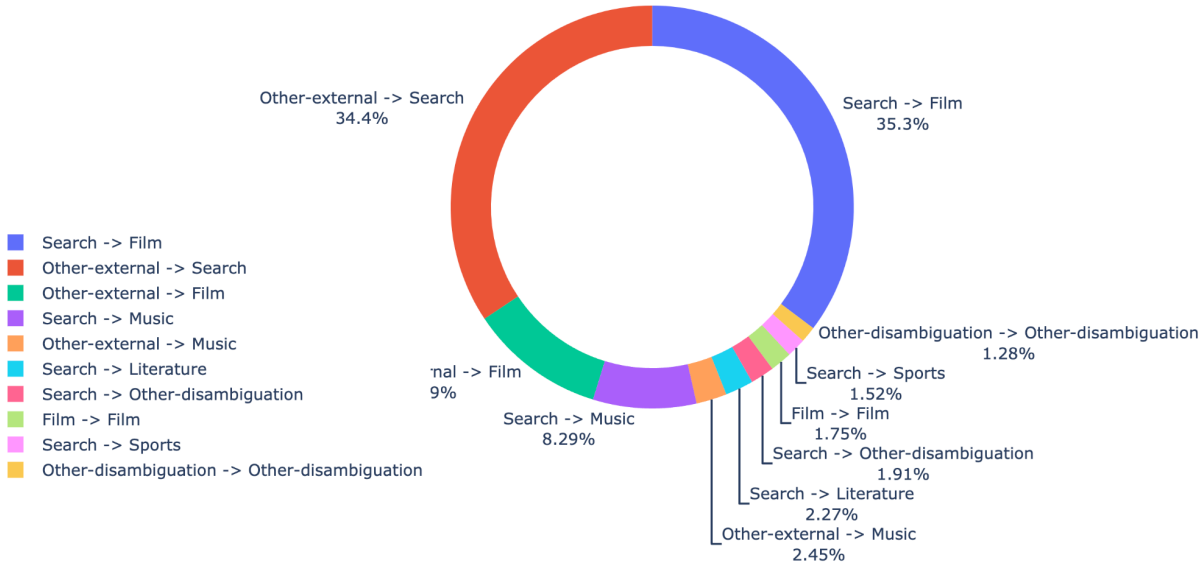


Exhibit showing donut charts for top 10 transitions, showing that ~35% of all transitions happen from a Search Engine to a Wikipedia Film article (excl. Other to Other transitions)

Originating Article -> Landing Article transitions



Predicting Film article with reasonable performance on precision / recall using Logistic Regression

Youdotcom Wikimedia Analysis doc

DP

---

1. We're able to predict if an article is a film based on previous article categories with reasonable performance
    a. We have 67% precision so among TP/TP+FP = 69% (among all predicted Film, we predicted 69% of them correctly)
    b. We have 81% recall so among TP/TP+FN 8/10 times we're predicted all of Films correctly,
    c. F-1 score which is the harmonic mean of precision and recall, we're at 74%

Interpreting results for predicting landing on "Film Articles" on Wikipedia

1. **`Search (Google, Bing, etc)`**: 0.1801 — This strong positive coefficient shows that if the previous article is categorized as a 'Search' (search engines or navigation pages). This is what we saw in correlation analysis as well.
2. **`Business`**: 5.352e-05 — This is a very small positive coefficient, meaning that if the previous article is in the 'Business' category,
3. **`Film`** : 0.0173This positive coefficient indicates that if the previous article is in the 'Film' category, it slightly increases the likelihood that the next article will also be in 'Film'
4. **`Other-external`**: -0.1806 — This negative coefficient is significant, meaning if the previous article is in 'Other-external' (likely external or miscellaneous content), it greatly decreases the likelihood that the next article will be in 'Film'.

Exhibit showing coefficients for various features that we engineered based on Wikipedia prior article categories

Coefficients for previous Article Categories predicting a "Film Article"

```
prev_Business: 5.35245881597101e-05
prev_Education: -1.6234997999897195e-05
prev_Entertainment: -6.763278224910709e-05
prev_Film: 0.01733543401904624
prev_Geography: -0.000905152951849518
prev_Health: -6.702704455477824e-06
prev_History: -2.4294485850290654e-05
prev_Literature: 0.0009160080945909427
prev_Music: -0.005217760029471664
prev_Other-disambiguation: -0.010313226213083494
prev_Other-external: -0.18064113141953117
prev_Politics: -3.350905614849994e-05
prev_Religion: -0.00027057936328343075
prev_Science: -0.00036879936427390163
prev_Search: 0.18018090167703135
prev_Sports: -0.000570535755521401
prev_Technology: -6.345910681447646e-05
```

Classification Report

Youdotcom Wikimedia Analysis doc

DP

```
Classification Report (Weighted):
                      precision    recall  f1-score   support

            Business       0.00      0.00      0.00 0.01028842126389789
           Education       0.00      0.00      0.00 0.0025603536085672244
       Entertainment       0.00      0.00      0.00 0.004191303754626686
                Film       0.67      0.81      0.74 0.49351773267830856
           Geography       0.00      0.00      0.00 0.013208342960775575
              Health       0.00      0.00      0.00 0.0011476876722362837
             History       0.00      0.00      0.00 0.0013017130273135769
          Literature       0.00      0.00      0.00 0.032661910748689175
               Music       0.00      0.00      0.00 0.10414401378106876
Other-disambiguation       0.00      0.00      0.00 0.033396677457521624
      Other-external       0.00      0.00      0.00 8.13215683302338e-06
            Politics       0.00      0.00      0.00 0.0012169344692365119
            Religion       0.00      0.00      0.00 0.008547407149041654
             Science       0.00      0.00      0.00 0.011695490169210156
              Search       0.28      0.98      0.43 0.058911039669582485
              Sports       0.00      0.00      0.00 0.019378962656806678
          Technology       0.00      0.00      0.00 0.007548896815309116

            accuracy                           0.57 0.803725020039025
           macro avg       0.06      0.11      0.07 0.803725020039025
        weighted avg       0.43      0.57      0.48 0.803725020039025
```

Blueprint for analysis

—

1. **Data Engineering**
    a. Set up a SQLite database for the english dump for a specific month (in this case December 2023's english dataset)
    b. Most easiest set up for a database within Python for us to work through
2. **Random Sampling**
    a. We sample the dataset to 10% to be able to easily process through pandas since we want to focus on the analysis. Our assumption is that 10% is representative of the database.
    b. If we find distribution similar, let's work with the sample, since it will cover larger trends in any case.

3. **Analysis:**
    a. Understand Summary Statistics
    b. Article level analysis (too granular, but we can eyeball the top articles users are navigating to)
        i. Understand top 10 articles previous articles users navigate from
        ii. Understand top 10 articles current articles users navigate to
        iii. Understand top 10 transitions from previous → current

Youdotcom Wikimedia Analysis doc

DP

---

c. **Map articles into predefined categories** (using a mapper)
   i. Understand the top 10 trends within each bucket
   ii. Understand concentration of transitions between categories

Exhibit showing the columns and their descriptions

Column	**Description**
prev	The title of the Wikipedia article the user visited before the current one, or a special value indicating the source type (example 'other-search' may mean other search engines like Google, external of wikipedia).
curr	The title of the current Wikipedia article the user visited
type	link : The user followed an internal link from another Wikipedia article. External: The user came from an external site (e.g., search engine). Other: Other types of referrer
n	The number of occurrences of this (prev, curr, type) combination during the month.

Prediction Task: Logistic Regression to predict the category of the next "Article" based on the previous article

1. In the 'prev' and 'curr' columns, find the top 50 occurring tokens and map them to category. Create a dictionary with category and mapping these tokens to specific categories
2. Used this category dictionary to map each article (in 'prev' to 'curr') to a category to make the prediction task more bounded and useful
   a. **[Manual]** Initially, had used a static list and was only able to map 4% of articles in 'curr', with tokenization, then sorting top 500 tokens and manually classifying it into categories, we're able to improve it 6.5% (+2.5%) (possible to automate this step)
3. Filtered for only mapped categories, removing "other" as its less useful
4. One hot encoded all "Wikipedia Categories" as inputs (e.g. Film, Search, Music, etc) to predict landing categories "Film, Music, etc". Encoded categories to numerical values to use logistic regression (which works on numerical inputs and assumes linearity between response and inputs)

Youdotcom Wikimedia Analysis doc

DP

---

5. Added weights to the prediction to account for the number of links, we fit **sample weights** to make sure that the transitions that occur more frequently have higher weights
6. The weight was normalized since we saw large values of “N” to not allow numerical anomalies
7. Used logistic regression to predict the output in terms of a confusion matrix
8. The confusion matrix had various performance parameters such as accuracy, precision and f1 score among others and we were able to get high precision/recall for predicting “Film articles” and were able to build out an equation to predict “Films” where the highest coefficient was that the previous article was “Search”
9. We faced insufficiency of prediction due to df being sampled at 10% and then included the entire data in df to train the model with more data