K-Means Clustering on Demographic Data

Yunlu Li and Alex Snow

Abstract

Currently, ComScore's total digital data is partitioned across a variety of classifiers. Often, entities within these classifications vary heavily in terms of their demographic breakdowns. In the interest of further understanding similarities between our data, we employed a machine learning clustering algorithm, with hopes of finding useful insight for future marketing analytics; specifically, for clients with a product and target demographic, looking to place their ads in optimal locations across the web.

Methods

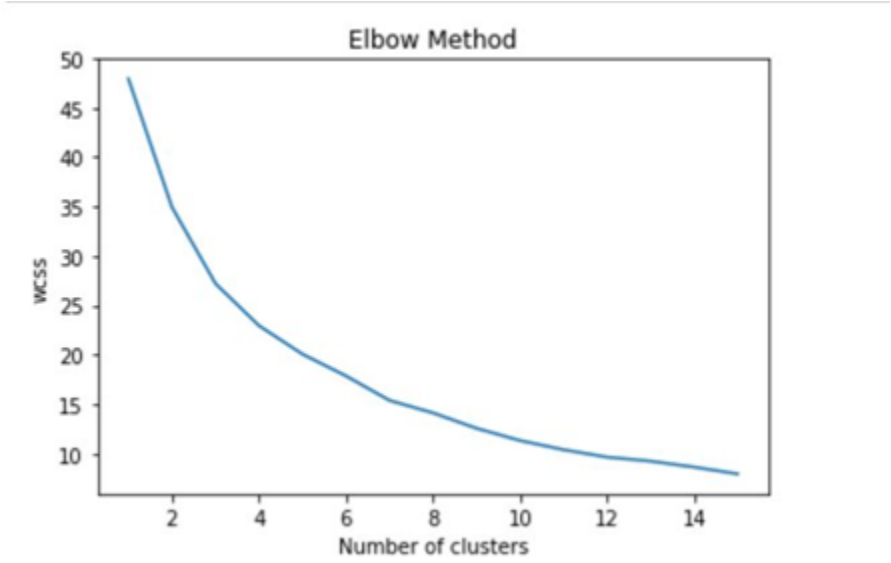
We believed the percentage metric for unique visitors for each demographic segment may be appropriate labels for understanding similarities in the entity's traffic patterns. Because of time constraints, and to avoid redundant classification, we focused only on the unique visitors percentage breakdown for the gender/age bucket. The implementation can easily be expanded to the other metrics and demographic buckets. However, running the same methodology on the household income bucket proved to be somewhat ineffective for a variety of reasons, one of which being the dominance of higher income households in nearly all the entities' data.

The data from month 230 (February 2019) were pivoted to achieve the following format (head of data frame is shown), allowing for each of the columns to serve as measures for the k-means clustering algorithm:

web_id	Female: 18-24 uv_pct	Female: 2-17 uv_pct	Female: 25-34 uv_pct	Female: 35-54 uv_pct	Female: 55+ uv_pct	Male: 18-24 uv_pct	Male: 2-17 uv_pct	Male: 25-34 uv_pct	Male: 35-54 uv_pct	Male: 55+ uv_pct
10429	6.734956	1.152563	10.057733	28.715391	5.009871	6.576584	1.258092	9.768232	25.289755	5.436823
10531	6.789288	0.481121	10.919898	37.679132	8.456660	2.964353	0.437999	7.642237	19.877530	4.751782
10583	2.869065	0.200320	9.631877	34.478493	6.140279	3.073528	0.142136	8.955290	28.347889	6.161124
10672	5.215922	1.441251	11.114770	30.837584	4.608127	4.668497	1.440910	8.505274	26.439640	5.728025
10779	3.607843	0.334328	11.166894	31.842077	6.369707	3.072440	0.252579	9.063088	27.847712	6.443531

The data were then preprocessed by standardizing the values along the measure axes, to avoid scaling issues.

This was an unsupervised learning process, and as such, we employed what is known as the “elbow” method, to help choose the optimal number of clusters (k-means requires specification of this parameter beforehand). The k-means algorithm was run on the scaled data in python for clusters of size 1 to 16, and the wcss (within cluster’s sum of squares) was calculated for each. This numeric represents the sum of the total squared deviation of each point within a cluster to the cluster’s centroid. This graph is shown below:



The “elbow” in the curve is supposed to be the point of interest, i.e. where the slope first begins to become significantly less severe. This point can be interpreted as the cutoff for where additional clusters become irrelevant, no longer significantly decreasing the wcss. This can usually be identified visually, although here it was not immediately obvious. However, this point can be found by drawing a straight line between the furthest left and right points and calculating the distance between this new line and each point on the wcss curve. The “elbow” is the point that maximizes this distance. We found this point to be at k = 4.

The data were then fit to the k-means model and used to generate 4 labels. The centroids of these k-means were placed into a table along with their weights for each measure. This weight can be used to qualitatively interpret the data, noting what scores are high relative to the other clusters. This table is shown below:

Out[25]:

	Female: 18-24 uv_pct	Female: 2-17 uv_pct	Female: 25-34 uv_pct	Female: 35-54 uv_pct	Female: 55+ uv_pct	Male: 18-24 uv_pct	Male: 2-17 uv_pct	Male: 25-34 uv_pct	Male: 35-54 uv_pct	Male: 55+ uv_pct
0	0.113070	0.062381	0.217872	0.553760	0.307284	0.159625	0.045022	0.255431	0.481437	0.320157
1	0.066752	0.009170	0.094057	0.190562	0.099015	0.318351	0.015169	0.469206	0.811686	0.464526
2	0.418193	0.154456	0.252282	0.252236	0.070813	0.550149	0.131985	0.437187	0.241201	0.062088
3	0.179526	0.020301	0.348508	0.743512	0.307449	0.106983	0.010798	0.189109	0.324885	0.184784

The index of the graph represents the label assigned by the k-means model. Based on this table, we assigned the following qualitative labels to each cluster:

0 – Professional (35+): Significantly high relative weightings for 35-54 age ranges and moderately high relative weightings for 55+ age ranges

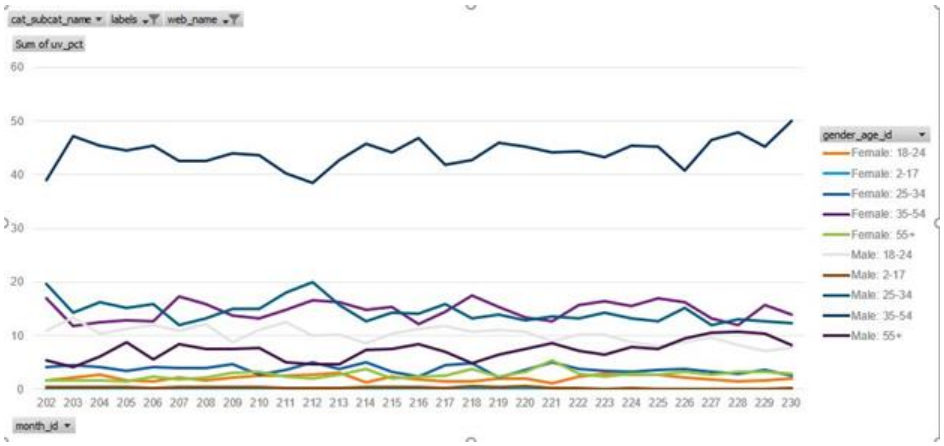
1 – Male: Significantly high relative weightings for all Males 25+

2 – Young (18-34): Significantly high relative weightings for 18-34 age ranges

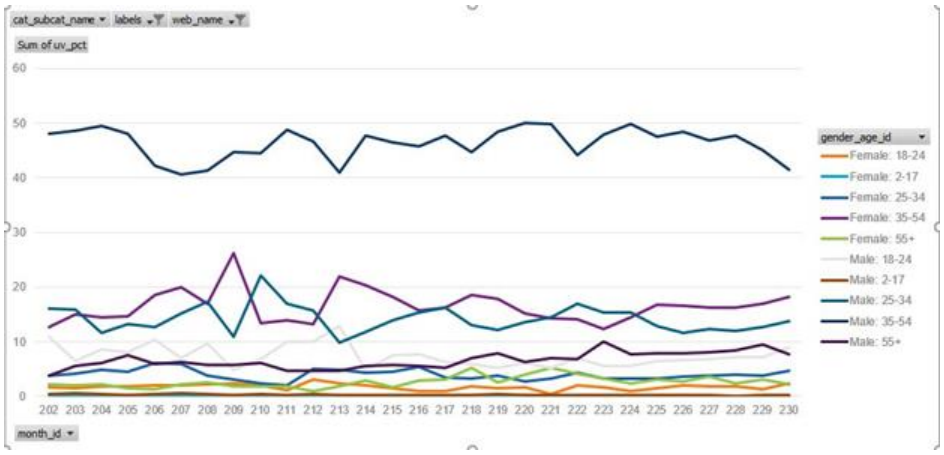
3 – Female: Significantly high relative weightings for all Females 25+

After further analyzing the data, entities within the same label appeared to follow the qualitative descriptions. A graph of two entities each from two different labels is given below:

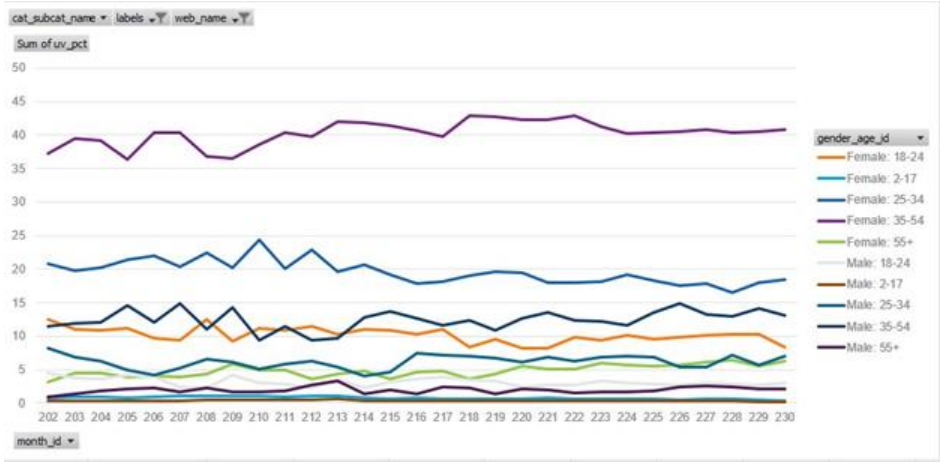
Bleacher Report Media (Male)



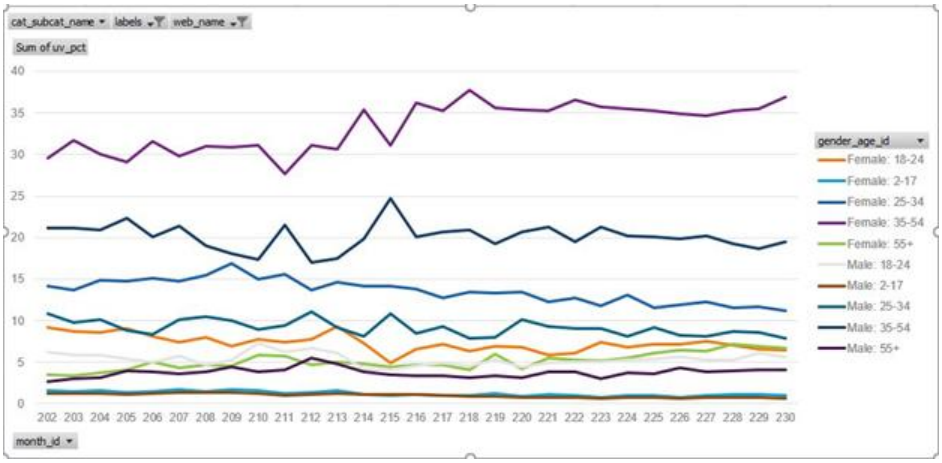
CBSSPORTS.COM (Male)



ETSY.COM (Female)



BLOGGER.COM (Female)



The BLOGGER.COM graph is a good example of the classification being relative to other distributions. ETSY's second highest segment for month 230 is indeed Male's 35-54 at 19.2%, but this is 5% lower than average for this segment, about 1 deviation from the mean among all those sampled. However, its Female 25-34 percentage is 36.9% during the same month, about a full deviation above the mean of 29%. The combination of these deviations is what constitutes the entity's final classification.

Conclusion:

We had originally explored the k-means algorithm out of curiosity, without setting our expectations too high for what we might find. After careful implementation, we were able to discover 4 clusters that captured a decent amount of the variation between entities' respective unique visitors percentage breakdowns for the gender age bucket. Although our implementation was not incredibly advanced, the sample was relatively large and representative of the top clients at ComScore, and our model delivered justifiable results for understanding hidden similarities within their data.

If anything, we hope this will help us to better understand similarities between client's data that aren't obvious from their current categorizations within our system. Ideally, expansion of this process will help to reveal more granular levels of insight, or more robust categorizations, helping clients and advertisers understand similarities between the traffic patterns of a variety of large web entities. This may also see its use in the quality assurance process - if the month over month categorization proves to be rather robust, those entities that switch labels for the current month may be flagged for further inspection.