Section 1 : Problem

Web data view is a very useful tool for facilitating extraction of information from various websites and optimizes the copy-paste operation. The operation is not only time consuming but the structure of the web pages make it extremely difficult to select the appropriate elements at times.

1.1

Scenario 1 : Civil Engineering Decision Making

Procurement of construction equipment is a major part of a Construction Engineer's job profile. The requirements for each project are different and a detailed analysis of the various buying/leasing options has to be performed to optimize construction costs.

Web data view can prove to be an invaluable tool in a situation like this as it can help in the creation of a tools database from various websites and these results can be processed together. The information can then be sorted to prioritize the user's requirement as well as graph the tradeoffs between various attributes like Engine Capacity, Cost, and Fuel efficiency to name a few. A number of decisions can be made based on the generated results. Some example are,

1. Buying/leasing/Rental
2. Fuel Efficiency vs Cost optimizing equipment
3. Transportation costs contributing to reduction in profit margin

A typical construction equipment website is shown in Fig. 1.1 along with some example clusters we will expect to see. The screenshot in Fig 1.2 indicates our final intended results.

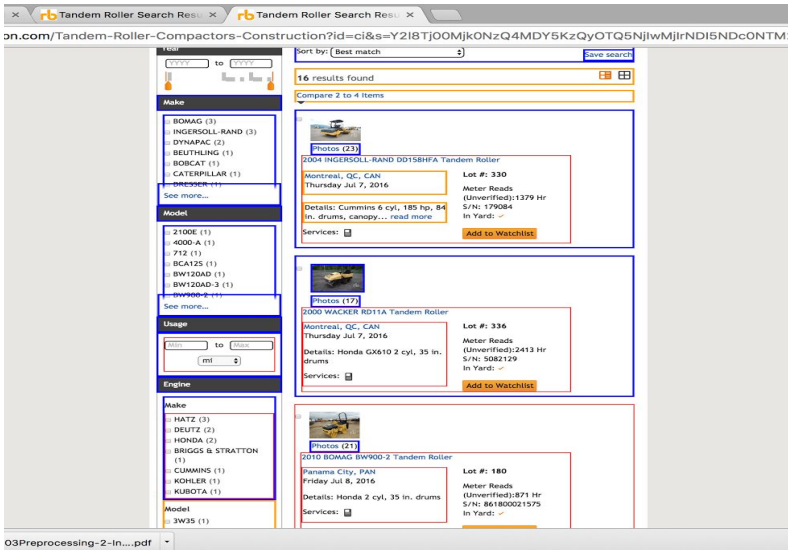


Fig 1.1 Typical Equipment Website

Vehicle	Location	Engine Details
2004 INGERSOLL-RAND DD158HFA Tandem Roller	Montreal, QC, CAN	Cummins 6 cyl, 185 hp, 84 in. drums, canopy
2000 WACKER RD11A Tandem Roller	Montreal, QC, CAN	Honda GX610 2 cyl, 35 in. drums
2010 BOMAG BW900-2 Tandem Roller	Panama City, PAN	Honda 2 cyl, 35 in. drums

Fig 1.2 Intended Excel Output

Scenario 2 : Life Cycle Analysis for John Deere

A U of I research project builds excel tools that analyzes data for John Deere equipment. Web data view (WDV) will potentially extract the input data and output various metrics that can generate a life cycle analysis of John Deere products and their predicted emissions.This information can aid John Deere in marketing their products as eco-friendly and ensure they are within emissions regulations. A typical situation is described below.

Let's consider we are evaluating a John Deere tractor. Some specifications are never mentioned as part of the marketing strategy and need to be obtained from the "Bill of Materials."In a normal situation , the user has to navigate to the John Deere webpage, find the missing part, and copy and paste the entire breakdown of things that contribute to that part. An example of the parts for a Hydrostatic Hose Clamping is shown in Fig 1.3. WDW can expedite the entire process.

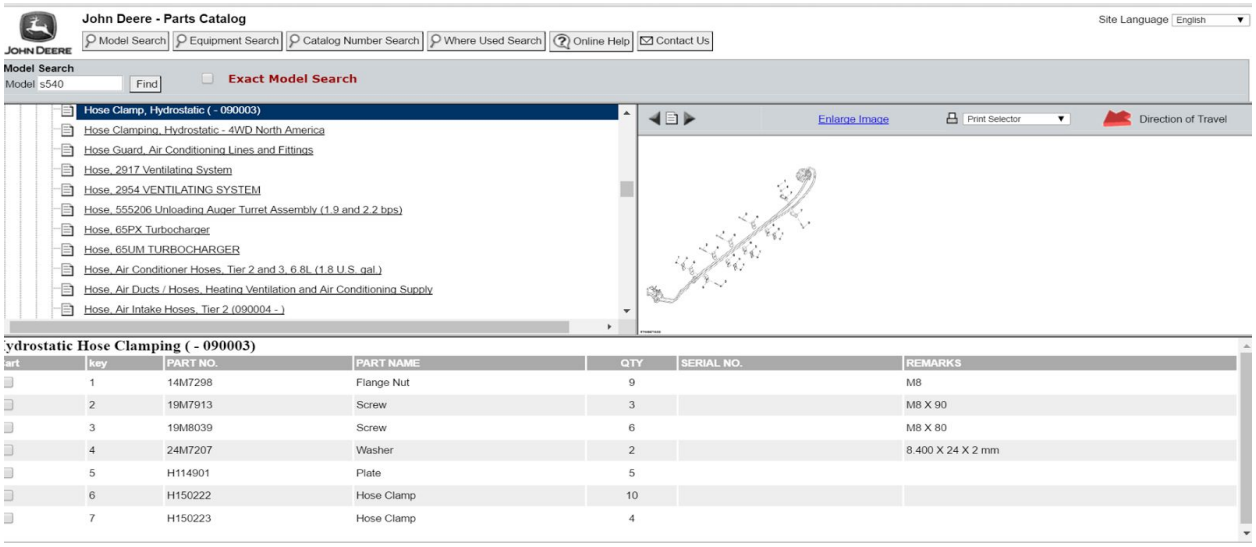


Fig 1.3 Example of Information Extraction for LCA

1.2
Problem Breakdown

Clustering/Classification

The problem can be approached from both directions, but we restricted our efforts to clustering for the following reasons,

1. Classification requires an extensive training set and time for generating meaningful results.
2. In order to expand our domain, classification seemed restrictive as different websites have very different type of attributes and class labels would change substantially.
3. The structure of commercial web pages is very uniform and is therefore more inherently suited to clustering.

Input/Output

We aim to require minimal input from the user. The user would click an element(s) on the webpage and the program would highlight all the corresponding clusters the elements belong to. We have not actively implemented the 'click' input from the user at this stage.

Domain

We extended our domain from equipment websites to any webpage that has a product/specification format e.g. common shopping websites with a structured visual representation. The structuring of the webpage plays an important role in our features selection as well as our choice of algorithm.

Section 2 : Solution

2.1

Features Selection

Initial Attempt and Final Selection

Complex web pages have several thousand attributes and many of do not have numeric values. Many of them are further branched and thus the overall process of integrating them becomes a challenging problem.

1. We eliminated all attributes that were neither string nor numerical as we didn't find any efficient way to include them
2. Next, we used one-hot encoding to convert all string attributes into numerical ones. This increased the data set significantly as one string attribute was converted into several numerical attributes.
3. We used Variance Threshold method for filtering attributes, but because of the principle of the algorithm (It eliminated attributes that had the same value across feature vectors) the data set was not reduced very significantly
4. The results obtained diluted the important attributes and we got very random results because a strong correlation was not found between the attributes being unique and the attributes being useful.

Final Selection

1. We observed that in our domain (e-commerce webpages) "Shape and location of one visual block usually can indicate its content or type."
2. When we used inherently numeric attributes (such as width, height, offset locations, start and end locations),our experiment results were much better, as these webpages are almost always structured.

2.2

Algorithm Selection

We initially attempted to create a Manual Model and subsequently progressed to K-Means and DBScan and Hierarchical techniques. The algorithms are enumerated in the order in which they were attempted,

Manual Model

1. The first thing we tried was to manually cluster certain features together based on key attributes
2. For example, the images would always have a png or jpeg tag.The title Font would usually be larger than a threshold value and would typically have certain style attributes etc.
3. We soon realized that this would limit us in expanding our domain as different web pages use very different specifications criteria and the method worked well only for the specific web pages we designed it for.

K-Means

Our next choice was K-means based on it's intuitive centroid based clustering approach but this did not work very well for our problem set. Figure 2.1 illustrates how similar features on the same webpage are assigned different clusters.

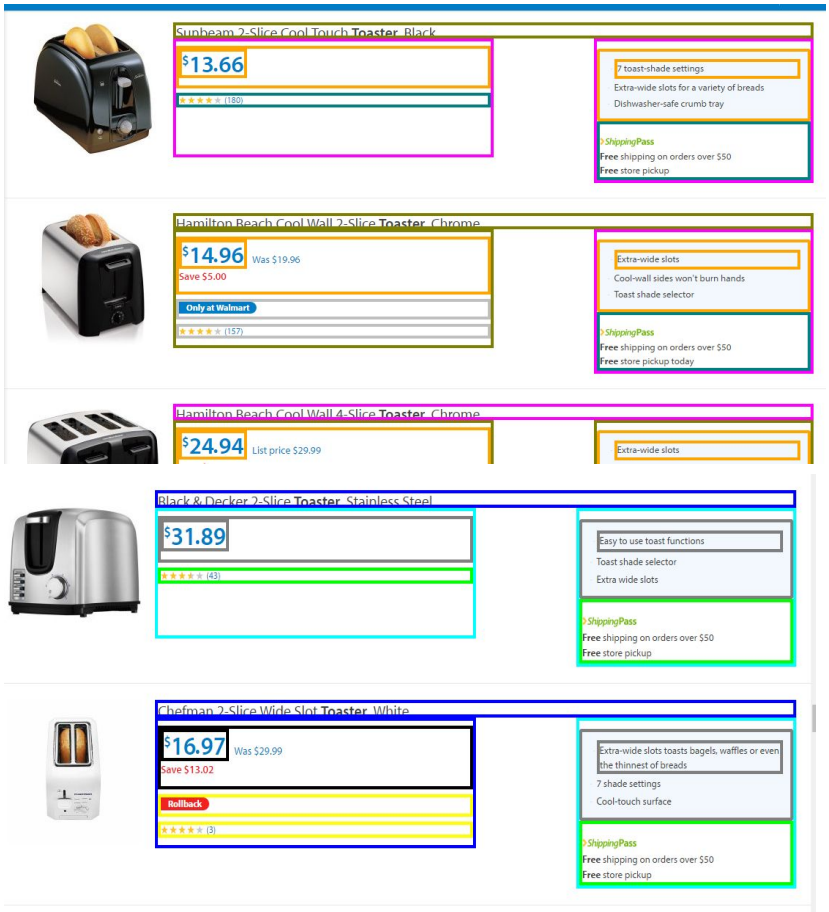


Fig 2.1 K-means Results (Recall Failure)

Reasons for Failure

The major failure of this algorithm was that while B-Cubed Precision was not very disappointing the recall was very low. A combination of several factors and assumptions may have contributed to this behaviour.

1. K-means chooses centroids to minimize inertia between clusters (within cluster sum of squares)
2. It has no concept of noise.
3. It is implicitly designed for Euclidean distance and is therefore did not provide us with the flexibility of choosing a distance measure.
4. Assumes clusters are convex and isotropic and it is possible our data is not convex. Figure 2.2 illustrates this point.
5. The K value differs based on the webpage and is a runtime parameter. An approximate user guess was often wrong because there is no way to insure that relevant data (and not noise) is more specifically clustered. Figure 2.3 illustrates this.

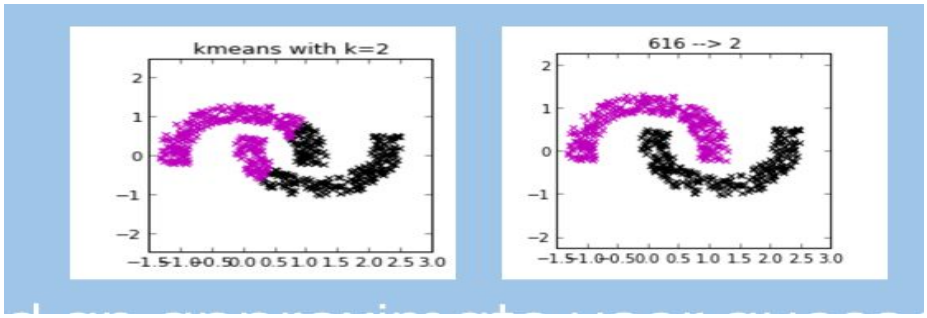


Fig. 2.2 K-means clustering for non convex data vs ground truth

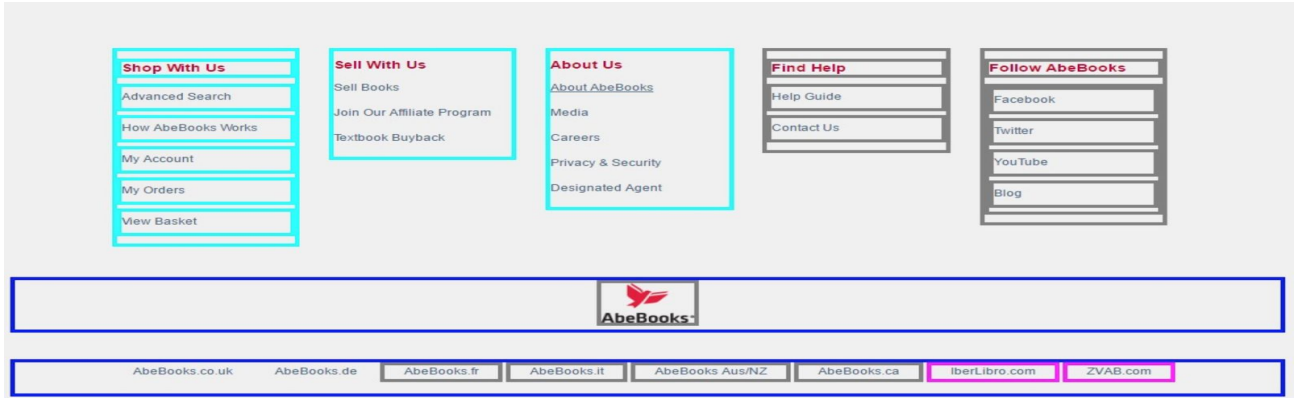


Fig 2.3 The noise being clustered instead of important attributes

DB Scan/Hierarchical

Motivation

We simultaneously experimented with both of these algorithms and moved forward with DBScan as the results were slightly better and we were making progress more rapidly.
DBScan is applied in order to improve the clustering performance. According to the previous experiments on K-means, visual blocks are more likely to be allocated into the same cluster based on their locations. Visual blocks that are close to each other in the webpage lead to a smaller difference in K-means's measure. This leads to the failure of clustering the same type of visual blocks that are located through the entire web page.

DBScan is density based. The algorithm views the clusters as high density areas separated by the low density. It requires two parameters as epsilon and min_samples, but number_clusters is not required anymore. DBScan first decides the set of core samples. A core sample is basically a data sample that has min_samples neighbours within the distance of epsilon. Then, clusters are generated based on recursively finding the core samples. Remaining points will be clustered as noise/outliers.

Reasons for Success

1. It has the concept of noise and can help eliminate outliers.
2. No run time parameters are required from the user thus minimizing user involvement.
3. Different clusters in our domain have approximately the same density because of the structure even though our data might not always be convex.
4. DB-scan fits for the structure of visual blocks' feature vector. Since the feature vectors are kind in the higher dimension, it is pretty inaccurate to directly measure their distance. But visual blocks of the same type will share several similar location and shape information, it makes these points stay close in most directions and generate a denser sample set.
5. Based on the domain knowledge of the visual blocks and some test runs, it shows that min_samples can be set as a certain number (5)Thus, we primarily tuned the epsilon parameter.

2.3
Parameter Setting and Algorithm options :

There are four main issues/choices,
1) Data normalization
2) Weight of the feature attributes
3) Parameter tuning
4) Distance functions.

Normalization of Data

The first issue is the normalization of the feature vectors. Since our feature attributes are all numeric, the procedure is not challenging. The common way to normalize is max-min normalization and zero mean unit variance normalization. We picked the max-min normalization since we don't want to have negative values in the normalized data. The main reason for the normalization is to get rid of the absolute values' bias. So that the cluster's can formulate the valid results. Also, the normalization itself is not enough since different attributes will have different weights. This leads to our next issue.

Weight of Attributes

Although based on our insight, location and shape information can represent a visual block's type well, the weight between them is not clear initially. During the test runs, usually if the weight is evenly distributed, it will works very successfully on the list style web pages. However, for the grid style web page, the visual blocks are more likely to be clustered by the location information. This usually leads to the clusters of grid style visual blocks to be a column. The main reason of such cluster result is that the location attributes have too much weight in the feature vector. Thus, we applied some scaling method to redistribute the weight of feature vectors. The solution comes in two parts. First, we still have to normalize the feature vectors along each attributes. Second, we scale down the location related feature attributes. It is proved to work very well when we scale down the location related features attributes to be 30~40% respecting to the shape related features.

Parameter Tuning

This is a common issue for almost every machine learning algorithms. And it usually can only be tuned manually to reach better performance. For example, K-means requires the selection of number of clusters which has to be done by reinitialize the algorithms with different K values and compare the average variance in the clusters. For DBScan, there are two parameters epsilon

and min_samples. However, based on the domain knowledge we have and the test runs , it shows that min_samples will not affect the performance of the clusters very much. Thus, we take a common value for the min_samples which is 5.

Once we fix the min_samples, there is a systematic way to choose the epsilon. As mentioned in Tan (2006), the basic approach is to calculate the k-distance plot which contains the distance from a point to its k^{th} nearest neighbor which k is the value of min_samples. The idea is that for the data points in the same cluster, the k-dist should be relatively small and for the noise points, the k-dist should be relatively large. Thus, if we draw the sorted k-dist plot, there will be an elbow point in the graph. And that corresponding distance can be used as the epsilon value.

In order to tune our DBScan model, one naive way is to do it for each website since usually the web pages from the same website will generate similar JSON files and will have similar cluster results for the same model. However, it is still very costly if we would like to apply this algorithm to general web pages. Thus, another way to adjust the parameters is to formulate a set of general parameters and then ask the algorithm to apply different pairs of parameters based on the information of the visual blocks. For example, if most visual blocks' widths are close to their heights, it is highly possible that the webpage is in grid style. Thus, it's data point density distribution will be different from the list style webpage. Thus, we can pick a certain pair of parameters based on our former experiments on those different styles webpages.

One issue inside our specific problem is that our results is more related with visualization and it usually has to be manually decided whether the performance is good or not if the truth data is not given. This leads to the inevitable human labor efforts. If a proper evaluation metric can be provided or labeled truth data is available, some tuning process can become automatic.

Distance Function

The choice of distance function is the last main issue. Since the location and the shape information are mainly geometric attributes. At first we thought that Euclidean distance may work better. However, after several test runs with Manhattan distance, it turns out that if we explicitly adding weights to certain attributes, the Euclidean distance works better. But for the general evenly distributed features, the Manhattan distance is better. The reason of such performance is basically if the weight is uneven, Euclidean distance will help higher weight attributes to distinguish from other attributes and more easily to form a denser area. However, in the even distributed case, the Manhattan distance will enlarge the distance comparing to the Euclidean distance. It helps to separate the data points.

Section 3 :Demo

Please refer to the demo videos for all the results obtained for DBScan algorithm and the presentation included with this report.

Video 1 : Successful Run with Amazon
Video 2 : Successful Run with Walmart
Video 3 : Before Tuning Epsilon
Video 4 : After Tuning Epsilon

Please note that we made progress on our tuning method from what was presented in class and is mentioned in the Parameters Tuning Section above.

Section 4:Evaluation

4.1

Failures/Challenges of DBScan

The results are meaningless if the core assumption of uniform variance is not fulfilled as shown in Figure 4.1.

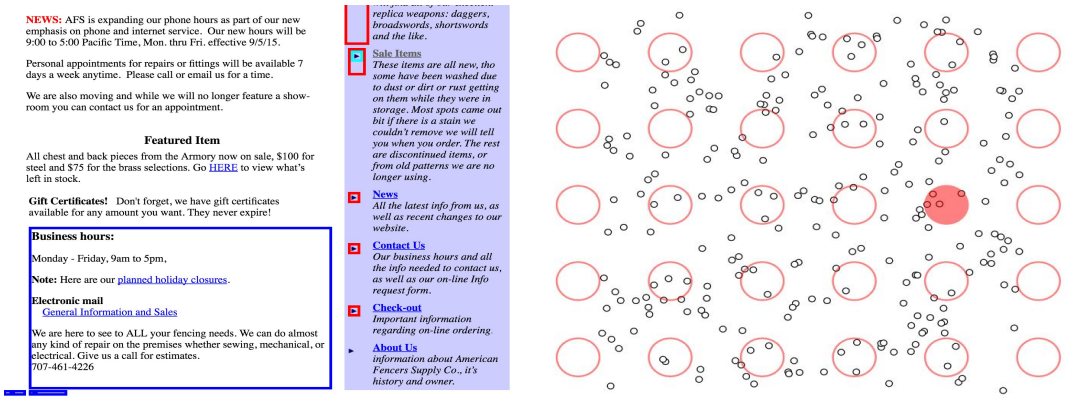


Fig 4.1 Failure of DBScan

Also, DBSCAN usually will not work if the webpage has a mixed layout style on its items, such as the left of the page is a list and the right of the page is a map (Airbnb-like webpage).

4.2

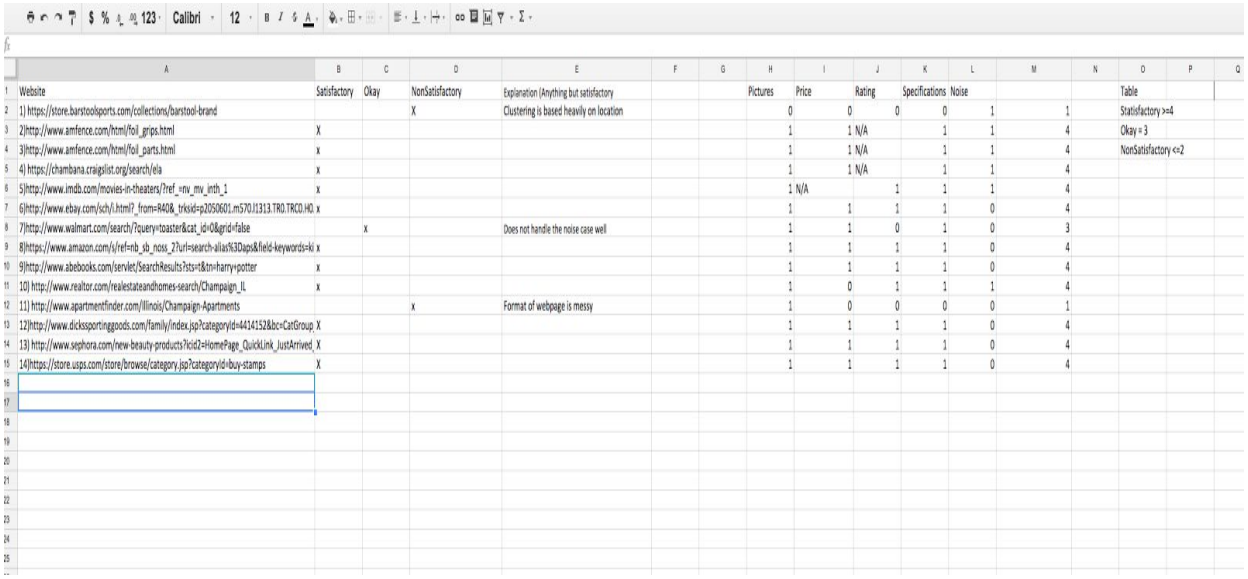
Metric

The scope of our web data view experience is restricted to online shopping. We decided on 5 typical categories based on what a person visiting the website would have to copy and paste manually,and assigned a score for each website. The grading categories are pictures, price, rating, specifications, and noise handling.

If the tool was able to correctly cluster the pictures, price, rating, or specifications, it was assigned a 1 on each criteria. The score was subsequently averaged from the 5 categories. If the result was above 75 percent, it was classified as satisfactory. If the result was between 60 and 75 percent, it was classified as fair. Finally, if the result was below 60 percent, it was classified as not satisfactory.

We found that the best results came from web pages with list formatting. When the format of the website not structured well, the tool had some issues with clustering the correct attributes together. A way to improve this would be to improve the selection of our attributes.

From our sample size, approximately 80 percent results that were satisfactory. 90 percent of the results were at least fair, so the method can be deemed satisfactory overall. The results are presented in Figure 4.2. (The spreadsheet is included separately).



	Website	Satisfactory	Okay	NonSatisfactory	Explanation (Anything but satisfactory			Pictures	Price	Rating	Specifications	Noise			Table
2	1) https://store.barstoolsports.com/collections/barstool-brand			X	Clustering is based heavily on location				0	0	0	1	1		Statisfactory >=4
3	2)http://www.amfence.com/html/foil_grips.html	X							1	1 N/A	1	1	4		Okay = 3
4	3)http://www.amfence.com/html/foil_parts.html	x							1	1 N/A	1	1	4		NonSatisfactory <=2
5	4) https://chambana.craigslist.org/search/ela	x							1	1 N/A	1	1	4		
6	5)http://www.imdb.com/movies-in-theaters/?ref_=nv_mv_inth_1	x							1 N/A	1	1	1	4		
7	6)http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR0.TRC0.H0.	x							1	1	1	1	0	4	
8	7)http://www.walmart.com/search/?query=toaster&cat_id=0&grid=false		x		Does not handle the noise case well				1	1	0	1	0	3	
9	8)https://www.amazon.com/s/ref=nb_sb_noss_2?url=search-alias%3Daps&field-keywords=ki	x							1	1	1	1	0	4	
10	9)http://www.abebooks.com/servlet/SearchResults?sts=t&tn=harry+potter	x							1	1	1	1	0	4	
11	10) http://www.realtor.com/realestateandhomes-search/Champaign_IL	x							1	0	1	1	1	4	
12	11) http://www.apartmentfinder.com/Illinois/Champaign-Apartments			x	Format of webpage is messy				1	0	0	0	0	1	
13	12)http://www.dickssportinggoods.com/family/index.jsp?categoryId=44141528&bc=CatGroup	X							1	1	1	1	0	4	
14	13) http://www.sephora.com/new-beauty-products?icid2=HomePage_QuickLink_JustArrived_	X							1	1	1	1	0	4	
15	14)https://store.usps.com/store/browse/category.jsp?categoryId=buy-stamps	X							1	1	1	1	0	4	

Fig 4.2 Evaluation based on custom metric

Section 5:Future Work

Some upgrade ideas are,

1. Upgrade the user experience. The first part of upgrade would be to add pictures and color to the actual chrome extension. We could let the user personalize their extension background.
2. Use more sophisticated algorithms like 'Optics' that minimize the parameters that need to be tuned.
3. Implement a more formal B-Cubed evaluation metric in our testing stage to determine the efficiency of our method.
4. Find a better way to extract feature information from the raw web page data that can differentiate between inside a single block id.
5. Include optional user feedback to update our statistical analysis for each web page.This would allow us to form a better success metric.
6. Upgrade our features vector. As we mentioned previously, we would like to add some non numerical attributes so that the tool can cluster websites with bad formatting.

References

1. Tan, P. N. (2006). *Introduction to data mining*. Pearson Education India.
2. Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.