International Journal of Computer Science and Engineering (IJCSE)
ISSN Print: 2278–9960; ISSN Online:2278–9979;
Vol. 8, Issue 6, Oct–Nov 2019; 9–20
© IASET

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations
IASET

A REVIEW ON FINDING TOP K-COMPETITOR FROM LARGE UNSTRUCTURED DATASETS BASED C-MINER ALGORITHM

J. Hemalatha[1] & M. Abhilash Kumar Reddy[2]

[1]PG Scholar, Department of Computer Science and Engineering, SV Engineering College for Women, Tirupati, Andhra Pradesh, India

[2]Assistant Professor, Department of Computer Science and Engineering, SV Engineering College for Women, Tirupati, Andhra Pradesh, India

ABSTRACT

In this new age any focused business and achievement is relying upon the capacity to make a thing more alluring to the client than the challenge. Some of the inquiries are coming in this undertaking, for example, first question is that: (I Who are principle contenders of the given things? II) How to formalize and evaluate intensity between things? Furthermore, III) What are various highlights of a thing that most influence its competitiveness?.In this Paper we validated both quantitatively and qualitatively. Our formalization is appropriate crosswise over spaces, beating the deficiencies of past methodologies.

To operationalize and address the issue of finding the top-k competitors of a thing in any given market. Especially within the sight of large data sets with hundreds or thousands of things, for example, those that are regularly found in standard spaces. We address these difficulties by means of a profoundly adaptable system for top-k computation, including a proficient assessment calculation and a suitable record.

KEYWORDS: *Data Mining, Web Mining, Information Search and Retrieval, Electronic Commerce*

Article History

Received: *04 Oct 2019* | ***Revised:*** *18 Oct 2019* | ***Accepted:*** *24 Oct 2019*

INTRODUCTION

A Long Line of research has shown the vital significance of distinguishing and checking an association's rivals. Spurred by this issue, the showcasing and the board network have concentrated on experimental strategies for contender distinguishing proof just as on techniques for breaking down known contenders. Extant research on the previous has concentrated on mining similar articulations (for example "Item A is better than Item B") from the Web or other sources. Despite the fact that such articulations can undoubtedly be pointers of intensity, they are missing in numerous spaces. For example, think about the vacation packages (for example flight-Stay Car Combination). For this situation, things have no appointed name by which they can be questioned or contrasted and one another.

Further, the recurrence of literary relative proof can shift enormously crosswise over spaces. For instance, when looking at brand names at the firm level (for example "Google Vs Yahoo" or "Sony Vs Panasonic"), almost certainly, near examples can be found by essentially questioning the web. Be that as it may, it is anything but difficult to recognize standard spaces where such proof is very rare, for example, hoes, jewellery, hotels, restaurants, and furniture. Motivated by these shortcomings, we propose a new formalization of the competitiveness between two items, based on the market segments that they can both cover.

Competitiveness

Our competitiveness paradigm is based on the following observation: the competitiveness between two items is based on whether they compete for the attention and business of the same groups of customers (for example a similar market sections). For instance, two eateries that exist in various nations are clearly not focused, since there is no cover between their objective gatherings.

This model shows the perfect situation, where we approach the total arrangement of clients in a given market, just as to explicit market portions and their necessities. By and by, in any case, such data isn't accessible. So as to defeat this, we portray a technique for processing every one of the sections in a given market dependent on mining huge audit informational indexes. This technique enables us to operationalize our meaning of intensity and address the issue of finding the top-k contenders of a thing in some random market. As we appear in our work, this issue presents noteworthy computational difficulties, particularly within the sight of enormous datasets with hundreds or thousands of things, for example, those that are regularly found in standard spaces. We address these difficulties by means of an exceptionally adaptable structure for top-k computation, including an effective assessment calculation and a fitting list.

Our work makes the following contributions:

- A formal definition of the competitiveness between two items, based on their appeal to the various customer segments in their market. Our approach overcomes the reliance of previous work on scarce comparative evidence mined from text.
- A formal methodology for the identification of the different types of customers in a given market, as well as for the estimation of the percentage of customers that belong to each type.
- A highly scalable framework for finding the top-k competitors of a given item in very large data sets.

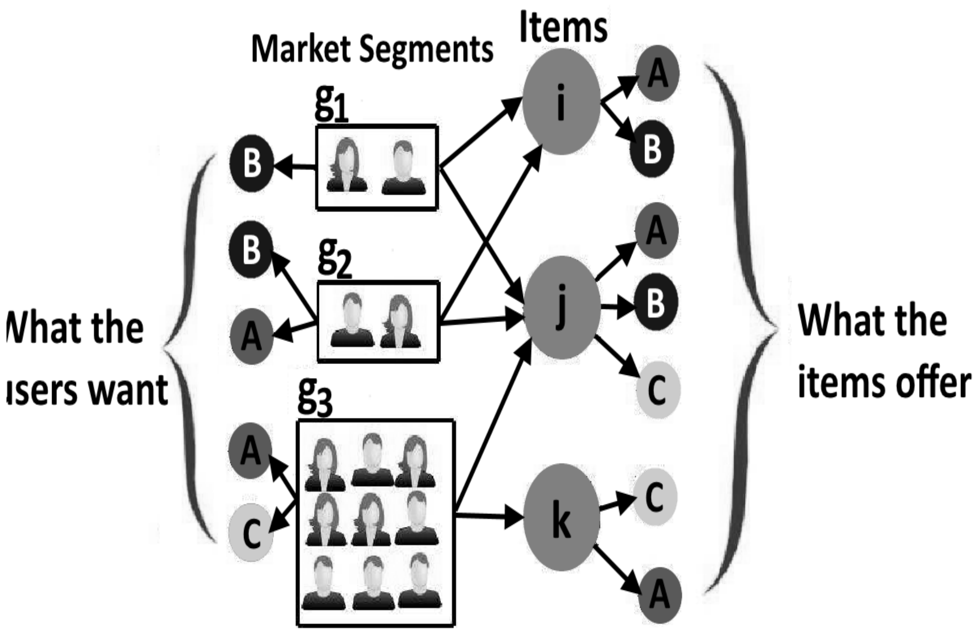


Figure 1: Competitiveness Paradigm.

LITERATURE SURVEY

The probabilistic dynamic skyline (PDS) question is an incredible asset for clients to use in choosing items as indicated by their inclinations. We detail a uncertain dynamic skyline (UDS) question over a probabilistic item set. Besides, we propose viable pruning methodologies for the UDS question, and coordinate them into powerful algorithms. The top k favourite probabilistic Products (TFPP) inquiry is displayed. The TFPP question is used to choose k items which can address the issues of a client set at the most extreme level. To handle the TFPP inquiry, we propose a TFPP calculation and its productive parallelization. Client inclinations data is a developing worry in market investigation. In this paper, we initially propose the UDS question to choose items that can fulfil a client's needs to the best degree.

Past research has inspected what drives administrative ID of contenders and how well administrators' apparent generally market structures coordinate client market structures. This examination tests the suggestion that how much a director precisely distinguishes contenders to their firm ought to improve firm execution. A company's general business experience showed an altered U association with precision. Regardless of whether the association's item hosted been guaranteed by a third-gathering endorser in the business was extensively identified with exactness, yet generally speaking had little association with execution. To make an upper hand and produce predominant execution, firms should initially recognize rivals. Nonetheless, there is small comprehension of how seen natural vulnerability influences contender distinguishing proof, why a few firms are better at recognizing local versus remote opponents, or how contender ID is identified with firm execution.

We present the after effects of a huge scale, start to finish human assessment of different estimation synopsis models. The assessment demonstrates that clients have a solid inclination for abridges that model notion over non-opinion baselines, yet have no expansive by and large inclination between any of the feeling based models. Be that as it may, an investigation of the human decisions recommends that there are recognizable circumstances where one abridges is by and large favoured over the others.

Proposed Algorithm

We propose another formalization of the aggressiveness between two things, in light of the market portions that the two of them can cover. We portray a technique for processing every one of the sections in a given market dependent on mining enormous survey datasets. This strategy enables us to operationalize our meaning of intensity and address the issue of finding the top contenders of a thing in some random market. As we appear in our work, this issue presents noteworthy computational difficulties, particularly within the sight of enormous datasets with hundreds or thousands of things, for example, those that are frequently found in standard areas. We address these difficulties by means of an exceptionally adaptable system for top calculation, including a productive assessment calculation and a fitting file.

Pair Wise Coverage

We begin by defining the pair wise coverage of a single feature f. We then define the pair wise coverage of an entire query of features q.

We define the pair wise coverage V^f of a feature f by two items i, jas the percentage of all possible values of f that can be covered by both i and j. Formally, given the set of all possible values V^f for f, we define:

$$V f_{i,j} = |\{v \in V f : v \angle f[i] \wedge v \angle f[j]\}|$$

|values(f)|,

where $v \angle f[i]$ represents that v is covered by the value of item i for feature f.

Next, we describe the computation of $V^f_{i,j}$ for different types of features.

Binary & Categorical Features

Unmitigated highlights take at least one qualities from a limited space. Instances of single-esteem highlights incorporate the brand of a computerized camera or the area of an eatery. Instances of multi-access highlights incorporate the luxuries offered by an hotel or the kinds of food offered by a restaurant. Any absolute element can be encoded by means of a lot of parallel highlights, with every paired element demonstrating the (absence of) inclusion of one of the first component's potential qualities. In this straightforward setting, the component can be completely covered (if $f[i] = f[j] = 1$ or, equivalently, $f[i] \times f[j] = 1$), or not covered at all. Formally, the pair wise coverage of a binary feature f by two items i, j can be computed as follows:

$$V^f_{i,j} = f[i] \times f[j] \text{ (binary features)}$$

Numeric Features

Numeric features take values from a pre-defined range. Henceforth, without loss of generality, we consider numeric features that take values in [0, 1], with higher values being preferable. The pair wise coverage of a numeric feature f by two items i and j can be easily computed as the smallest (worst) value achieved for f by either item. For instance, consider two restaurants i, j with values 0.8 and 0.5 for the feature food quality. Their pair wise coverage in this setting is 0.5. Conceptually, the two items will compete for any customer who accepts a quality ≤ 0.5. Customers with higher standards would eliminate restaurant j, which will never have a chance to compete for their business. Formally, the pair wise coverage of a numeric feature f by two items i, j can be computed as follows:

$$V^f_{i,j} = \min(f[i], f[j]) \text{ (numeric features)}$$

The C Miner Algorithm

Next, we present C Miner, an exact algorithm for finding the top-k competitors of a given thing. Our calculation utilizes the horizon pyramid so as to lessen the quantity of things that should be considered. Given that we just care about the top-k contenders, we can steadily figure the score of every competitor and stop when it is ensured that the top-k have developed. The pseudo code is surrendered.

Algorithm 1

Discussion of C Miner: The input includes the set of items I, the set of features F, the item of interest i, the number k of top competitors to retrieve, the set Q of queries and their probabilities, and the skyline pyramid DI. The algorithm first retrieves the items that dominate i, via masters (i) (line 1). These items have the maximum possible competitiveness with i. If at least k such items exist, we report those and conclude (lines 2–4). Otherwise, we add them to Top K and decrement our budget of k accordingly (line 5). The variable LB maintains the lowest lower bound from the current top-k set (line 6) and is used to prune candidates. In line 7, we initialize the set of candidates X as the union of items in the first layer of the pyramid and the set of items dominated by those already in the Top K. This is achieved via calling GETSLAVES (TopK, DI). In every iteration of lines 8–17, C Miner feeds the set of candidates X to the UPDATETOPK () routine, which prunes items based on the LB threshold. It then updates the TopK set via the MERGE () function, which identifies the items with the highest competitiveness from TopK. This can be achieved in linear time, since both and TopK are sorted. In line 13, the pruning threshold LB is set to the worst (lowest) score among the new TopK. Finally, GETSLAVES () is used to expand the set of candidates by including items that are dominated by those in X.

Algorithm 1 C Miner

Input: Set of items *I*, Item of interest $i \in I$, feature space *F*, Collection $Q \in 2F$ of queries with non-zero weights, skyline pyramid *DI*, in*k.*

Output: Set of top-*k* competitors for *i*

```
• $TopK \leftarrow masters(i)$
• if ($k \leq |TopK|$) then
• return $TopK$
• end if
• $k \leftarrow k - |TopK|$
• $LB \leftarrow -1$
• $X \leftarrow$GETSLAVES($TopK; DI$)$\cup DI[0]$
• while ($|X| != 0$ ) do
• $X$ UPDATETOPK($k; LB; X$)
• if ($|X| != 0$ ) then
• $TopK$ MERGE($TopK; X$)
• if ($|Top\ K| = k$) then
• $LB$ WORSTIN ($TopK$)
• end if
• $X$ GETSLAVES ($X; DI$)
• end if
• end while
• return $TopK$
• Routine UPDATETOPK ($k, LB, X$)
• $localTopK \leftarrow \emptyset$
• $low(j) \leftarrow \Sigma 0; \forall j \in X.$
• $up(j) \leftarrow p(q)\ X\ Vi;jq\ ; \forall j \in X.q2Q$
• for every $q \in Q$ do
• $maxV \leftarrow p(q)\ X\ Vi;iq$
• for every item$j \in X$ do
• $up(j) \leftarrow up(j) - maxV + p(q)\ X\ Vi;jq$
• if ($up(j) < LB$) then
• $X \leftarrow X \setminus \{j\}$
• else
• $low(j) \leftarrow low(j) + p(q)\ X\ Vi;jq$
• local TopK: update (j; low(j))
• if ($|local\ TopK| \geq k$ ) then
• $LB \leftarrow$ WORSTIN ($localTopK$)
• end if
• end if
• end for
• if ($|X| \leq k$) then
• break
• end if
• end for
• for every item $j \in X$ do
• for every remaining $q \in Q$ do
• $low(j) \leftarrow low(j) + p(q)\ X\ Vi;jq$
• end for
• local TopK: update(j; low(j))
• end for
• return TOPK ($localTopK$)
```

DISCUSSIONS OF UPDATETOPK ()

This normal procedures the up-and-comers in X and finds all things considered k applicants with the most noteworthy aggressiveness with I. The routine uses an information structure neighbourhood TopK, actualized as an acquainted array: the score of every competitor fills in as the key, while its id fills in as the worth. The exhibit is key-arranged, to encourage the calculation of the k best things. The structure is consequently truncated with the goal that it generally contains all things considered k things.

In lines 21–22 we introduce the lower and upper limits. For each thing j ∈ X, low (j) keeps up the present aggressiveness score of j as new inquiries are considered, and fills in as a lower bound to the up-and-comer's genuine score. Each lower bound low (j) begins from 0, and after the culmination of UPDATETOPK (), it incorporates the genuine aggressiveness score CF (I, j) of competitor j with the central thing I. Then again, up (j) is an idealistic upper bound on j's intensity score. At first, up (j) is set to the most extreme conceivable score (line 22). This is equivalent to ∑ q∈Q p (q) × V q i,i, where V Q i,i is just the inclusion given solely by I to q.

It is then incrementally reduced toward the true CF (i, j) value as follows. For every query q ∈ Q, maxV holds the maximum possible competitiveness between item i and any other item for that query, which is in fact the coverage of i with respect to q. Then, for each candidate j ∈ X, we subtract maxV from up (j) and then add to it the actual competitiveness between i and j for query q. If the upper bound up(j) of a candidate j becomes lower than the pruning threshold LB, then j can be safely disqualified (lines 27–29).

Otherwise, low (j) is updated and j remains in consideration (lines 30–31). After each update, the value of LB is set to the worst score in local TopK (lines 32–33), to employ stricter pruning in future iterations. If the number of candidates |X| becomes less or equal to k (line 37), the loop over the queries comes to a halt. This is an early-stopping criterion: since our goal is to retrieve the best k candidates in X, having |X| < = k means that all remaining candidates should be returned. In lines 41–46 we complete the competitiveness computation of the remaining candidates and update local Topk accordingly. This takes place after the completion of the first loop, in order to avoid unnecessary bound-checking and improve performance.

Administrator Module

In this module, an admin can upload details about items i.e. Camera, Hotels, Restaurants, and Recipes. After that, admin can checks all uploaded items details, customer queries and interests. Finally Topk competitors are identified from given item based on C Miner.

Customer Module

In the Second module, we develop the Customer based features. In this module, the customer can give queries for anyone item, i.e. Camera, Hotels, Restaurants and recipes. At first creating the data set for cameras, Hotels, Restaurant, Recipes. Collect the Customer requirement from customer page

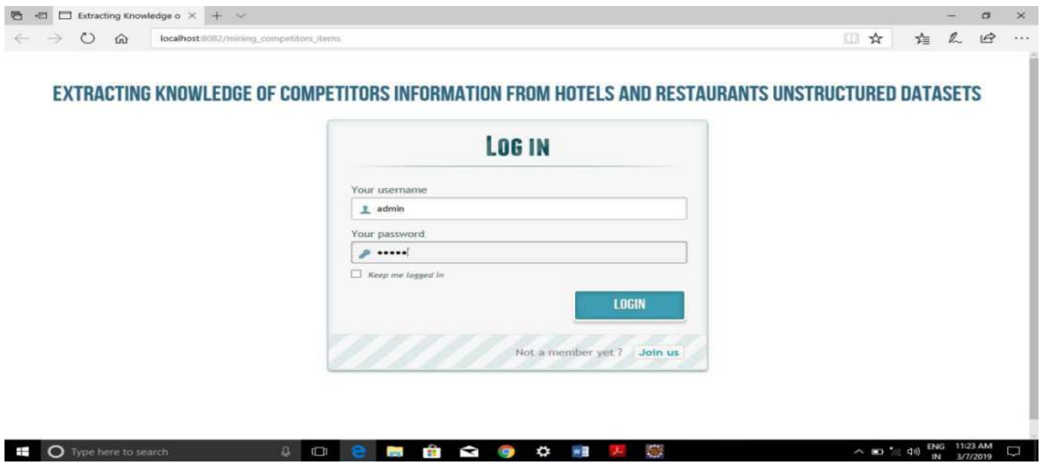
C Miner Algorithm Module

Next, we present C Miner, an exact algorithm for finding the top-k competitors of a given item. Our algorithm makes use of the skyline pyramid in order to reduce the number of items that need to be considered. Given that we only care about the top-k competitors, we can incrementally compute the score of each candidate and stop when it is guaranteed that the top-k has emerged.

Skyline Operator Module

In this module, skyline operator is performed. The skyline is a wells studied concept that represents the subset of points in a population that are not nominated by any other point. We refer to the skyline of a set of items i as Sky (I).The concept of the skyline leads to the following lemma: Lemma1. Given the skyline Sky(I) of a set of items I and an item i E I, let Y contain the k items from Sky(I)that are most competitive with i. Then, an item j E I can only be in the top-k competitors of i, if j E Y or if j is dominated by one of the items in Y.

RESULTS



EXTRACTING KNOWLEDGE OF COMPETITORS INFORMATION FROM HOTELS AND RESTAURANTS UNSTRUCTURED DATASETS

LOG IN

Your username: admin

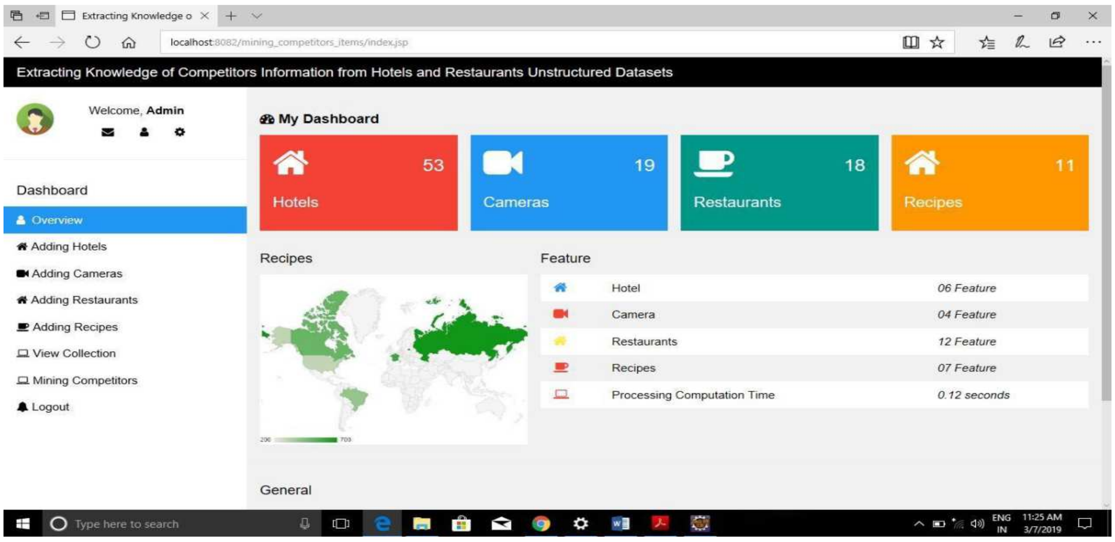
Your password: •••••

Keep me logged in

LOGIN

Not a member yet ? Join us

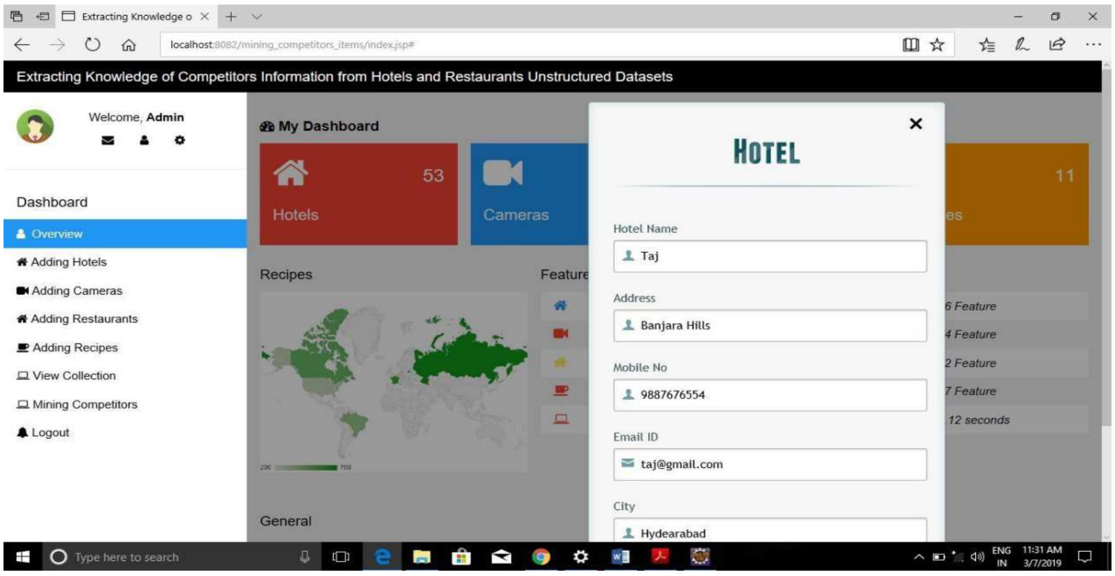
Figure 2: Login Page.



Feature

Hotel	06 Feature
Camera	04 Feature
Restaurants	12 Feature
Recipes	07 Feature
Processing Computation Time	0.12 seconds

Figure 3: Home Page.



HOTEL

Hotel Name: Taj

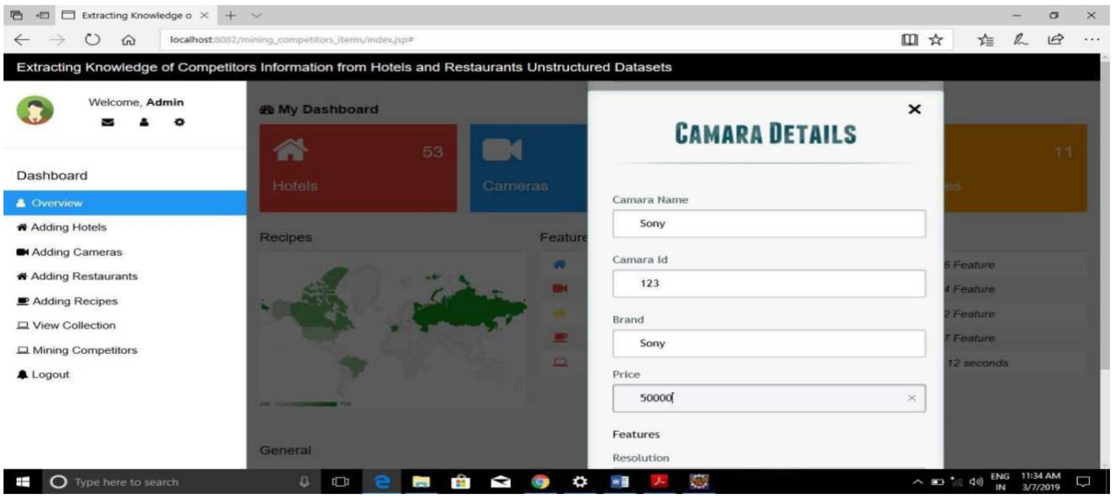
Address: Banjara Hills

Mobile No: 9887676554

Email ID: taj@gmail.com

City: Hydearabad

Figure 4: Adding Hotel.



CAMARA DETAILS

Camara Name: Sony

Camara Id: 123

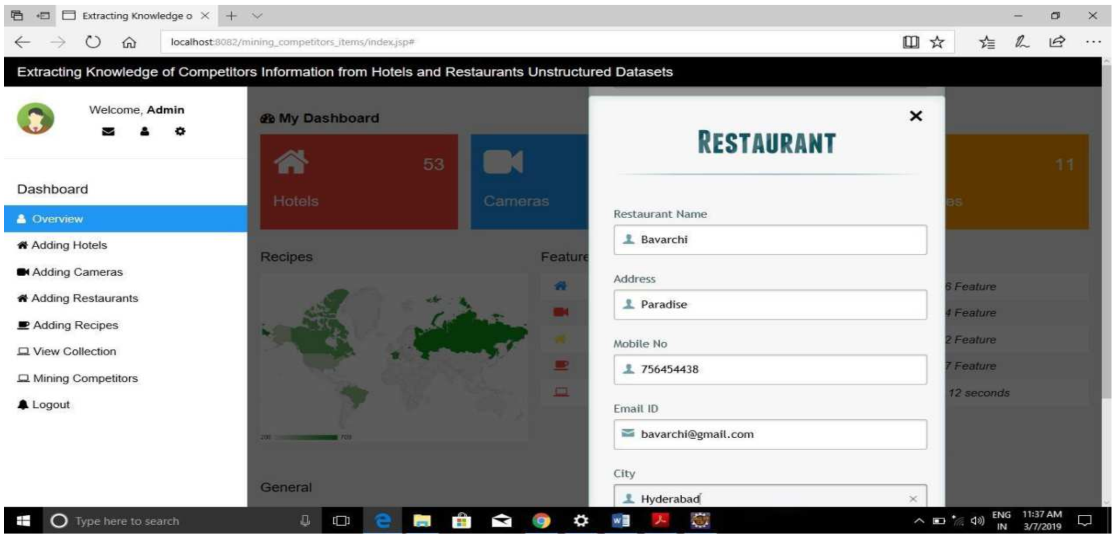
Brand: Sony

Price: 50000

Features

Resolution

Figure 5: Adding Camera.



Extracting Knowledge of Competitors Information from Hotels and Restaurants Unstructured Datasets

RESTAURANT

Restaurant Name: Bavarchi

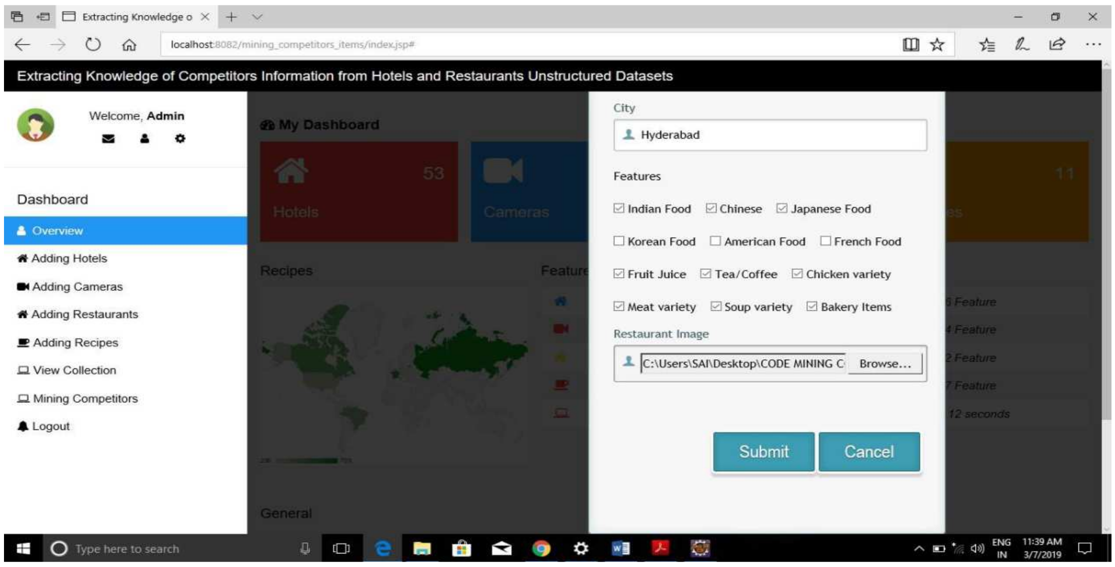
Address: Paradise

Mobile No: 756454438

Email ID: bavarchi@gmail.com

City: Hyderabad

Figure 6: Adding Restaurant.



Extracting Knowledge of Competitors Information from Hotels and Restaurants Unstructured Datasets

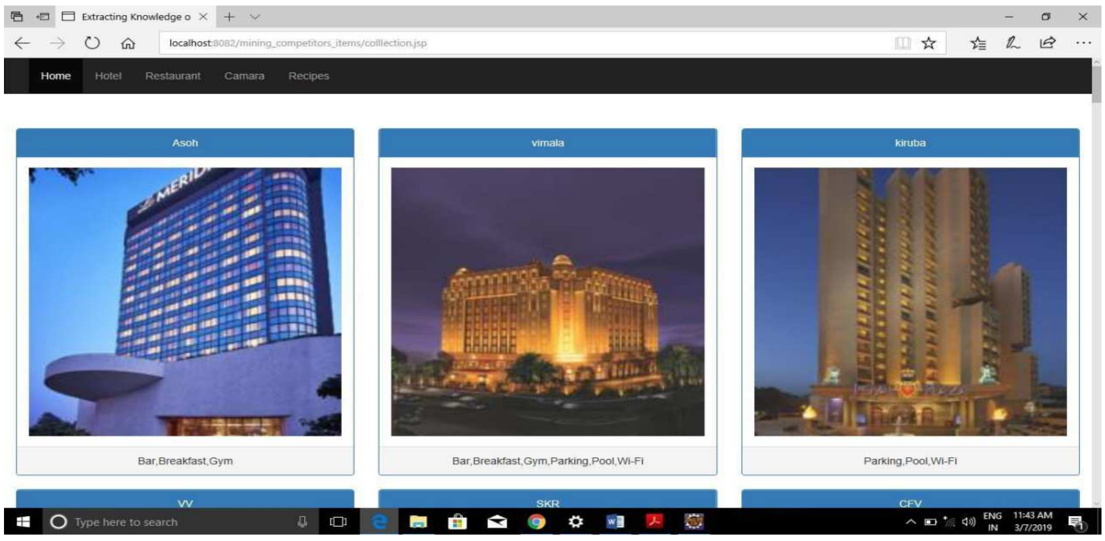
City: Hyderabad

Features

[✓ Indian Food] [✓ Chinese] [✓ Japanese Food]
Korean Food, American Food, French Food
[✓ Fruit Juice] [✓ Tea/Coffee] [✓ Chicken variety]
[✓ Meat variety] [✓ Soup variety] [✓ Bakery Items]

Restaurant Image: C:\Users\SAI\Desktop\CODE MINING C

Figure 7: Adding Recipes.



Asoh	vimala	kiruba
Bar,Breakfast,Gym	Bar,Breakfast,Gym,Parking,Pool,Wi-Fi	Parking,Pool,Wi-Fi
VV	SKR	CFV

Figure 8: View Collection.

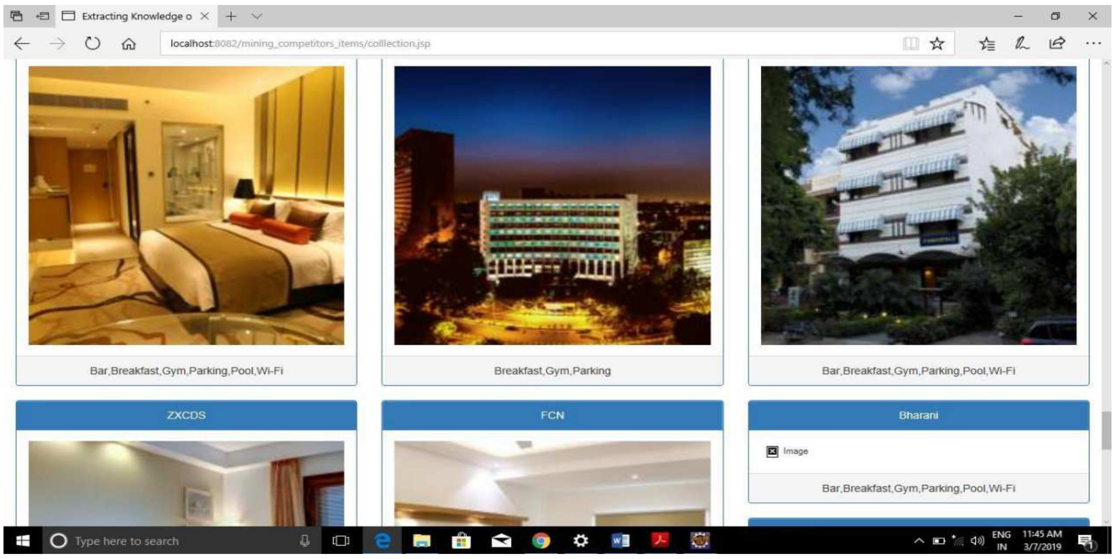

Extracting Knowledge o
localhost:8082/mining_competitors_items/colllection.jsp
Bar,Breakfast,Gym,Parking,Pool,Wi-Fi
Breakfast,Gym,Parking
Bar,Breakfast,Gym,Parking,Pool,Wi-Fi
ZXCDS
FCN
Bharani
Image
Bar,Breakfast,Gym,Parking,Pool,Wi-Fi
Type here to search
11:45 AM 3/7/2019

Figure 9: View Hotels.

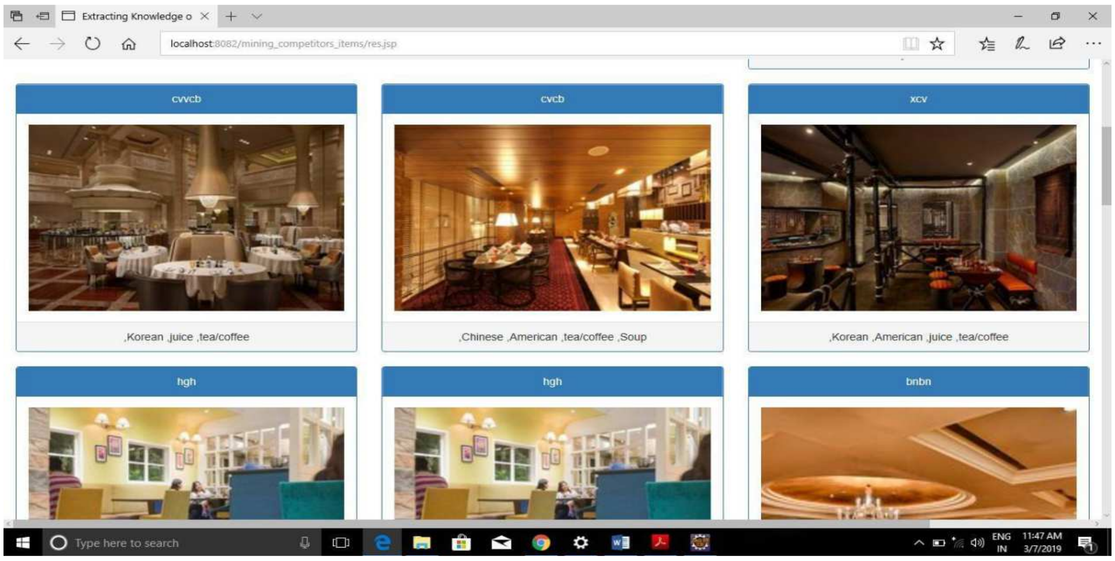

Extracting Knowledge o
localhost:8082/mining_competitors_items/res.jsp
cvvcb
cvcb
xcv
,Korean ,juice ,tea/coffee
,Chinese ,American ,tea/coffee ,Soup
,Korean ,American ,juice ,tea/coffee
hgh
hgh
bnbn
Type here to search
11:47 AM 3/7/2019

Figure 10: View Restaurants.

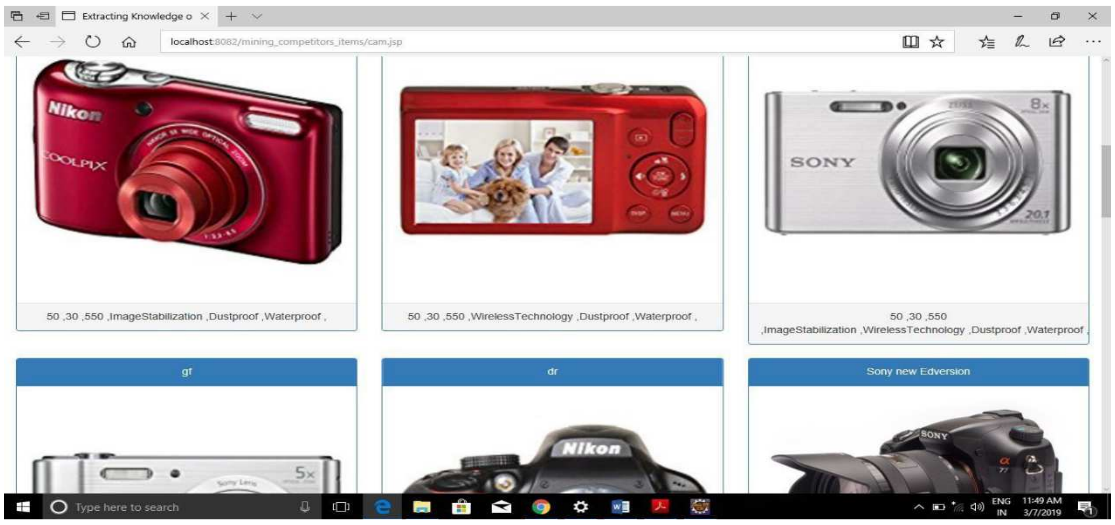

Extracting Knowledge o
localhost:8082/mining_competitors_items/cam.jsp
50 ,30 ,550 ,ImageStabilization ,Dustproof ,Waterproof ,
50 ,30 ,550 ,WirelessTechnology ,Dustproof ,Waterproof ,
50 ,30 ,550 ,ImageStabilization ,WirelessTechnology ,Dustproof ,Waterproof ,
gf
dr
Sony new Edversion
Type here to search
11:49 AM 3/7/2019

Figure 11: View Cameras.

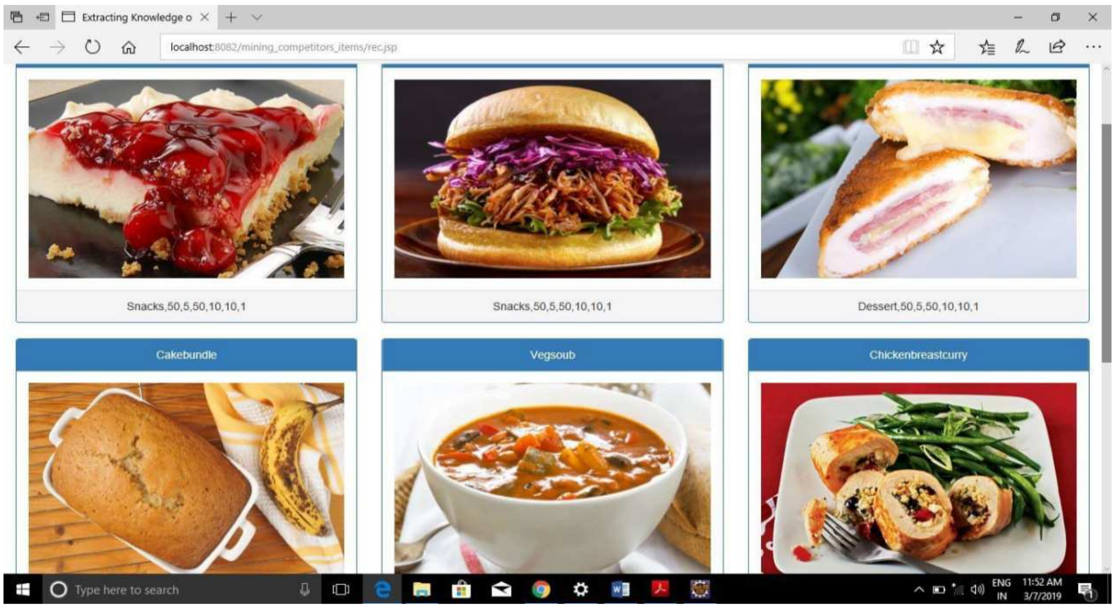


Figure 12: View Recipes.

CONCLUSIONS AND FUTURE WORK

We exhibited a conventional meaning of aggressiveness between two things, which we approved both quantitatively and subjectively. Our formalization is relevant crosswise over spaces, beating the weaknesses of past methodologies. We consider various components that have been to a great extent disregarded before, for example, the situation of the things in the multi-dimensional element space and the inclinations and assessments of the clients. Our work acquaints an end-with end philosophy for mining such data from huge datasets. In view of our aggressiveness definition, we tended to the computationally testing issue of finding the top-k contenders of a given thing. The proposed structure is effective and appropriate to spaces with extremely enormous populaces of things. The proficiency of our philosophy was confirmed by means of a trial assessment on genuine datasets from various spaces. Our examinations likewise uncovered that solitary few surveys is adequate to unquestionably evaluate the various kinds of clients in a given market, also the quantity of clients that have a place with each sort.

REFERENCES

1. *M. E. Porter, Competitive Strategy: Techniques for Analyzing Industries and Competitors. Free Press, 1980.*
2. *R. Deshpand and H. Gatingon, "Competitive analysis," Marketing Letters, 1994.*
3. *B. H. Clark and D. B. Montgomery, "Managerial Identification of Competitors," Journal of Marketing, 1999.*
4. *W. T. Few, "Managerial competitor identification: Integrating the categorization, economic and organizational identity perspectives," Doctoral Dissertaion, 2007.*
5. *M. Bergen and M. A. Peteraf, "Competitor identification and competitor analysis: a broad-based managerial approach," Managerial and Decision Economics, 2002.*
6. *J. F. Porac and H. Thomas, "Taxonomic mental models in competitor definition," The Academy of Management Review, 2008.*

7. *M. J. Chen, “Competitor analysis and inter firm rivalry: Toward a theoretical integration,” Academy of Management Review, 1996.*

8. *R. Li, S. Bao, J. Wang, Y. Yu, and Y. Cao, “Cominer: An effective algorithm for mining competitors from the web,” in ICDM, 2006.*

9. *Z. Ma, G. Pant, and O. R. L. Sheng, “Mining competitor relationships from online news: A network-based approach,” Electronic Commerce Research and Applications, 2011.*

10. *R. Li, S. Bao, J. Wang, Y. Liu, and Y. Yu, “Web scale competitor discovery using mutual information,” in ADMA, 2006.*