JCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-7, July 2018 E-ISSN: 2347-2693

Efficient Algorithms for Mining Top-K High Utility Itemsets

Ameena Aiman[1], Raafiya Gulmeher[2]

[1] M. Tech Student, KBNCE, Kalaburagi
[2] CSE Dept., KBNCE, Kalaburagi

Available online at: www.ijcseonline.org

Accepted: 20/Jul/2018, Published: 31/Jul/2018

Abstract- High utility itemsets (HUIs) mining is a developing topic in information mining, which alludes to finding all itemsets having a utility meeting a user-specified minimum utility threshold min_util. However, setting min_util appropriately is a difficult problem for users. Finding an appropriate minimum utility threshold by trial and error is a tedious process for users. If min_util is set too low, too many HUIs will be generated, which may cause the mining process to be very inefficient. On the other hand, if min_util is set too high, it is likely that no HUIs will be found. In this paper, we address the above issues by proposing a new framework for top-k high utility itemset mining, where k is the desired number of HUIs to be mined. Two types of efficient algorithms named TKU (mining Top-K Utility itemset) and TKO (mining Top-K utility itemset in One phase) are proposed for mining such itemset without the need to set min_util. We provide a structural comparison of the two algorithms with discussions on their advantages and limitations. Empirical evaluations on both real and synthetic datasets show that the performance of the proposed algorithms is close to that of the optimal case of state-of-the-art utility mining algorithms.

Keywords**—**ItemSets, Mining, High Utility, TKO, HUIs

I. INTRODUCTION

Today, the data has turned into a superabundant in all divisions, for example, from business exchanges and logical information, to satellite pictures, content reports and military insight. Even though it brings gigantic new advantages yet, it additionally makes enormous cerebral pain. Along these lines, it is critical to deal with the immense informational indexes in a viable way. Henceforth, Data mining is the way that aides in finding important examples from substantial informational indexes. In basic words, it is the way toward investigating information from alternate points of view and outlining it into valuable data. The general objective of the information mining process is to extricate data from an informational collection and change it into a reasonable structure for additionally utilize. Beside the crude investigation step, it includes database and information administration angles, information pre-preparing, model and induction contemplations, intriguing quality measurements, intricacy contemplations, post-handling of found structures, representation, and internet refreshing. In fact, the objective is the extraction of examples and learning from vast measure of information, not simply the extraction of information. Information mining is basically utilized today by organizations with a solid buyer center - retail, land, monetary exchanges, keeping money, media communications and showcasing associations. It empowers these organizations to decide connections among "interior" factors, for example, value, item situating, or staff abilities, and "outside" variables, for example, monetary pointers, rivalry, and client socioeconomics. What's more, it empowers them to decide the effect on deals, consumer loyalty, and corporate benefits. At long last, it empowers them to "penetrate down" into outline data to see detail value-based information. Finding helpful examples covered up in a database assumes a basic part in a few information mining undertakings, for example, high utility example mining. Mining high utility itemsets from a value-based database alludes to the revelation of itemsets with high utility like benefits. The accompanying portrayal brings the significance of utility mining which defeats the disadvantages of successive itemset and weighted affiliation lead mining. Visit designs [12] are itemsets, which show up in an informational collection with recurrence no not as much as a client indicated limit. For instance, an arrangement of things, for example, drain and bread that show up every now and again together in an exchange informational collection is a regular itemsets. Finding successive examples assumes a fundamental part in mining affiliations, connections, and numerous other intriguing connections among information. Visit design mining helps in information ordering, grouping, bunching, and other information mining assignments, for example, mining affiliation rules [3]. The uses of this sort of mining are in the field of broadcast communications, content examination and statistics investigation. The intriguing quality metric used to express the recurrence of itemsets is as far as help estimation of the itemsets. The Support estimation of an itemset is the level of exchanges that contain the itemset. Issue: In the structure of incessant itemset mining, the significance of things to clients isn't considered. Weighted affiliation [8] administer mining approach is an augmentation of the customary affiliation control mining [3] to which the weights are allocated to the things considering their essentialness. A weight of a thing is a non-negative genuine number that demonstrates the significance of everything. A couple (x, w) is known as a weighted thing where $x \in I$ is a thing and $w \in W$ is the weight related with x. An exchange is an arrangement of weighted things, every one of which may show up in numerous exchanges with various weights. Issue: Assigning weight for everything, considering its importance and creating the affiliation decides for vast itemsets that have over the client determined least weighted certainty. The constraints of regular or uncommon itemset mining inspired to build up a utility-based mining [9] approach, which enables a client to advantageously express his or her points of view concerning the convenience of itemsets as utility qualities and afterward find itemsets with high utility qualities higher than a client determined limit. In utility-based mining, the term utility alludes to the quantitative portrayal of client inclination i.e., the utility estimation of an itemset is the estimation of the significance of that itemset in the client's point of view. For e.g. on the off chance that a business expert engaged with some retail investigate necessities to discover which itemsets in the stores gain the most extreme deals income for the stores he or she will characterize the utility of any itemset as the fiscal benefit that the store procures by offering every unit of that itemset. Formally a thing set S is helpful to a client if it fulfills a utility limitation i.e. any limitation in the shape u(S) >= minutil, is helpful to a client where u(S) is the utility estimation of the itemset and minutil is a utility edge characterized by the client. An itemset is known as a high utility itemset if its utility is no not as much as a client determined limit; generally, the itemset is known as a low utility itemset. Existing strategies [3, 5, and 7] regularly produce a gigantic arrangement of high utility itemsets and the mining execution is debased therefore. The situation when the database contains numerous long exchanges will be the most exceedingly terrible. The gigantic number of potential high utility itemsets shapes a testing issue to the mining execution since the higher handling cost is acquired with more number of potential high utility itemsets. In this way, the principle point of our proposed calculation is to diminish the quantity of competitor itemsets. To accomplish this, we propose a proficient calculation, called UP-Growth (Utility Pattern Growth) with a minimized information structure called UP-Tree for finding high utility itemsets.

II. RELATED WORK

A. Definitions

Here we are examining some fundamental definitions about utility of a thing, utility of itemset in exchange, utility of itemset in database, related works and the issue associated with the current strategies. Given a limited arrangement of things I = {i1, i2, …. im} everything is $(1 \leq p \leq m)$ has a unit benefit pr (ip) An itemset X is an arrangement of k unmistakable things {i1, i2, …. ik}, where k is the length of X. An itemset with length k is known as a k-itemset. An exchange database D = {T1, T2…, Tn} contains an arrangement of exchanges, and every exchange Td has a novel identifier d, called TID. Everything ip in exchange Td is related with an amount q (ip, Td) that is, the acquired amount of ip in Td.

Definition 1: Utility of a thing ip in an exchange Td is the result of unit benefit and the bought amount. It is indicated as u (ip, Td) and characterized as pr (ip, Td) ×q (ip, Td)

Definition 2: Utility of an itemset X in Td is the whole of utilities of the exchange containing X. It is indicated as U (X, Td) and characterized as $\sum ip \mathcal{E} X \wedge X \subseteq Td$ u (ip, Td)

Definition 3: Utility of an itemset X in D is the entirety of utilities of the considerable number of exchanges containing X in the database D. It is signified as u(X) and $\sum X \subseteq Td \wedge Td \mathcal{E} D$ u (X, Td).

Definition 4: An itemset is known as a high utility itemset if its utility is no not as much as a client indicated least utility edge or else it is low-utility itemset.

Definition 5: Transaction utility of an exchange Td is the entirety of utilities of the considerable number of things in the exchange Td. It is signified as TU (Td) and characterized as u (Td, Td)

Definition 6: Transaction-weighted utility of an itemset X is the aggregate of the exchange utilities of the considerable number of exchanges containing X, which is meant as TWU(X) and characterized as $\sum X \subseteq Td \wedge Td \mathcal{E} D$ TU (Td)

Definition 7: An itemset X is known as a high-exchange weighted utility itemset (HTWUI) if TWU(X) is no not as much as least utility edge.

B. Literature Survey

The paper [3] proposed Apriority calculation, is utilized to acquire visit itemsets from the database. This calculation basically checks thing events to additionally decide the vast one itemsets which requires examining of database each time for everything. Next, the database examine is performed to tally the help of applicant's itemsets. At that point the affiliation rules are produced from visit itemsets. In the wake of distinguishing the huge itemsets, just those itemsets which have the help more noteworthy than the base help are permitted. Disadvantage: It creates a considerable measure of competitor thing sets and outputs database each time and when another exchange is added to the database then it ought to rescan the whole database once more.

The paper [5] proposed a productive FP-tree based mining technique that assembles visit design tree (FP-tree) structure, for putting away significant data about regular examples into compacted structure. Example piece development mines the total arrangement of regular examples utilizing the FP-development. It builds a profoundly minimal FP-tree, which is generally generously littler than the first database, by which expensive database examines are spared in the consequent mining forms. It applies an example development technique which stays away from exorbitant hopeful age. Downside: FP-Growth Consumes more memory and performs gravely with long example informational indexes. In this manner, it can't discover high utility itemsets. The paper [6] proposed Two-Phase calculation to proficiently prune down the quantity of hopefuls and can correctly acquire the total arrangement of high utility itemsets. In Phase I, High exchange weighted usage itemsets (HTWUIs) are recognized. The span of hopeful set is diminished by just considering the supersets of high exchange weighted usage itemsets. In Phase II, one database examine is performed to sift through the high exchange weighted usage itemsets that are for sure low utility itemsets. Disadvantage: It creates various contender to get HTWUIs and requires numerous database filters. Customary strategies for affiliation run mining think about the presence of a thing in an exchange, regardless of whether it is bought, as a twofold factor.

In any case, clients may buy more than one of a similar thing, and the unit cost may shift among things. Subsequently, building up an effective calculation is urgent for utility mining. To defeat this issue, the paper [7] proposed secluded things disposing of methodology (IIDS) to lessen the quantity of applicants. By pruning detached things amid level savvy seek, the quantity of applicant itemsets for HTWUIs in stage one can be decreased. Disadvantage: This calculation still outputs database for a few times and uses a hopeful age and-test plan to discover high utility itemsets and hence can't enhanced execution. The paper [2] proposed a tree-based calculation, named Incremental High Utility Pattern (IHUP). A tree-based structure called IHUP-Tree is utilized to keep up the data about itemsets and their utilities. Disadvantage: Though it accomplishes a superior execution than IIDS and Two-Phase, despite everything it creates a few HTWUIs. Since the overestimated utility figured by TWU is too substantial.

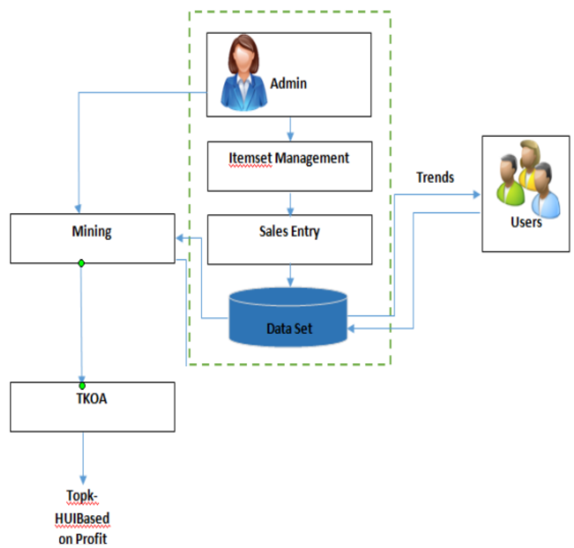
C. Problem Statement

In the literature survey we have examined the distinctive techniques proposed for high utility mining from substantial datasets. Be that as it may, this strategy every now and again creates a gigantic arrangement of PHUIs and their mining execution is debased therefore. Promote in the event of long exchanges in dataset or low edges are set, at that point this condition may turn out to be most noticeably bad. The gigantic number of PHUIs shapes a testing issue to the mining execution since the more PHUIs the calculation produces, the higher handling time it expends. Subsequently, to defeat these difficulties the proficient calculation is displayed in this paper. The fundamental point of this task is to accomplish the accompanying angles:

- Reducing the quantity of outputs in the first database.
- Minimize memory use (Reducing the inquiry space).
- Reducing the aggregate execution and calculation time
- Reducing the asset use.
- Increase the execution regarding time and space unpredictability

III. SYSTEM ANALYSIS

SYSTEM ARCHITECTURE:



EXISTING SYSTEM:

- The customary FIM (Frequent itemset mining) may find a lot of incessant yet low-esteem itemsets and lose the data on significant itemsets having low offering frequencies. Thus, it can't fulfill the necessity of clients who want to find itemsets with high utilities, for example, high benefits.
- To address these issues, utility mining develops as a vital point in information mining and has gotten broad consideration as of late. In utility mining, everything is related with a utility (e.g. unit benefit) and an event check in every exchange (e.g. amount).
- The utility of an itemset speaks to its significance, which can be estimated as far as weight, esteem, amount or

other data relying upon the client determination. An itemset is called high utility itemset (HUI) if its utility is no not as much as a client determined least utility limit min_util.

- In ongoing years, high utility itemset mining has gotten heaps of consideration and numerous productive calculations have been proposed, for example, Two-Phase, IHUP, IIDS, UPGrowth, d2HUP and HUI-Miner. These calculations can be for the most part arranged into two sorts: two-stage and one-stage calculations.

DISADVANTAGES OF EXISTING SYSTEM:

- ❖ Although many studies have been devoted to HUI mining, it is difficult for users to choose an appropriate minimum utility threshold in practice.
- ❖ The existing studies may perform well in some applications, they are not developed for top-k high utility itemset mining and still suffer from the subtle problem of setting appropriate thresholds.

PROPOSED SYSTEM:

In this paper, we address most of the above difficulties by proposing a novel structure for top-k high utility itemset mining, where k is the coveted number of HUIs to be mined.

Major contributions of this work are summarized as follows:

- First, two proficient calculations named TKU (mining Top-K Utility itemsets) and TKO (mining Top-K utility itemsets in One stage) are proposed for mining the total arrangement of best k HUIs in databases without the need to determine the min_util limit.
- The TKU calculation embraces a minimized tree-based structure named UP-Tree to keep up the data of exchanges and utilities of itemsets. TKU acquires valuable properties from the TWU model and comprises of two stages.
- In stage I, potential best k high utility itemsets (PKHUIs) are produced. In stage II, top-k HUIs are distinguished from the arrangement of PKHUIs found in stage I. Then again, the TKO calculation utilizes a rundown based structure named utility-rundown to store the utility data of itemsets in the database.
- It utilizes vertical information portrayal methods to find top-k HUIs in just a single stage.

ADVANTAGES OF PROPOSED SYSTEM:

- Two productive calculations TKU (mining Top-K Utility itemsets) and TKO (mining Top-K utility itemsets in One stage) are proposed for mining such itemsets without setting least utility limits.
- TKO is the first stage calculation produced for top-k HUI mining, which coordinates the novel procedures RUC, RUZ and EPB to enormously enhance its execution.
- Empirical assessments on various kinds of genuine and engineered datasets demonstrate that the proposed calculations have great versatility on vast datasets and the execution of the proposed calculations is near the ideal instance of the best in class two-stage and one-stage utility mining calculations.

IV. ALGORITHM

FHM(Fast High Utility Mining):

Our proposition depends on the perception that although HUI-Miner plays out a solitary stage and consequently don't produce competitors according to the meaning of the two-stage demonstrate, HUI-Miner investigates the inquiry space of itemsets by creating itemsets and an expensive join activity must be performed to assess the utility of each itemset. To diminish the quantity of joins that are performed, we propose a novel pruning procedure named EUCP (Estimated Utility Cooccurrence Pruning) that can prune itemsets without performing joins. This system is anything but difficult to actualize and exceptionally successful. We name the proposed calculation fusing this methodology FHM (Fast High-utility Miner). We analyze the execution of FHM and HUI-Miner on four genuine datasets. Results demonstrate that FHM performs up to 95 % less join activities than HUI-Miner and is up to six times speedier than HUI-Miner.

The FHM calculation:

In this segment, we show our proposition, the FHM calculation. The principle technique (Algorithm 1) takes as info an exchange database with utility qualities and the minutil edge. The calculation first sweeps the database to ascertain the TWU of everything. At that point, the calculation recognizes the set I* of all things having a TWU no not exactly minutil (different things are overlooked since they can't be a piece of a high-utility itemsets by Property 3). The TWU estimations of things are then used to set up an aggregate request ¬ on things, which is the request of climbing TWU esteems (as proposed in [7]). A second database examine is then performed. Amid this database check, things in exchanges are reordered by the aggregate request ¬, the utility-rundown of everything I? I * is fabricated and our novel structure named EUCS (Estimated Utility Co-Occurrence Structure) is manufactured. This last structure is characterized as an arrangement of triples of the shape (a, b, c)? I * × I * × R. A triple (a, b, c) demonstrates that TWU ({a, b}) = c. The EUCS can be actualized as a triangular grid or as a HashMap of HashMap's where just tuples of the shape (a, b, c) with the end goal that c 6= 0 are kept. In our usage, we have utilized this last portrayal to be more memory effective claiming we have watched that couple of things co-happen with different things. Building the EUCS (Estimated Utility Co-Occurrence Structure) is quick (it is performed with a solitary database check) and involves a little measure of memory, limited by |I * | × |I* |, despite the fact that practically speaking the size is significantly littler in light of the fact that a set number of sets of things co-happens in exchanges (cf. area 5). After the development of the EUCS, the profundity first hunt investigation of itemsets begins by calling the recursive technique Search with the void itemset Ø, the arrangement of single things I *, minutil and the EUCS structure.

ALGORITHM STEPS:

Algorithm 1: The FHM algorithm

Input : D: a transaction database, minutil: a user-specified threshold

Output: the set of high-utility itemsets

Step:1 Scan D to calculate the TWU of single items;

Step:2 I * ? each item i such that TWU(i) < minutil;

Step:3 Let be the total order of TWU ascending values on I * ;

Step:4 Scan D to build the utility-list of each item i ? * and build the EUCS structure;

Step:5 Search (Ø, I * , minutil, EUCS);

V. IMPLEMENTATION

MODULES

- List Utility-Items
- Frequent Itemset Mining
- Top-k Pattern Mining
- TKO (mining Top-K utility itemsets in One phase)

MODULES DESCRIPTION

List Utility-Items:

In this module, we build up the List Utility-Items by presenting the utility-list structure and related properties. For insights about utility-records, in the TKO Base and TKO calculations, each item(set) is related with a utility-list. The utility-arrangements of things are called introductory utility-records, which can be developed by examining the database twice. In the principal database examine, the TWU and utility estimations of things are figured. Amid the second database check, things in every exchange are arranged by TWU esteems and the utility-rundown of everything is built, where things in every exchange are organized in climbing request of TWU esteems. The utility-rundown of a thing (set) X comprises of at least one tuples. Each tuple speaks to the data of X in an exchange T r and has three fields: Tid, iutil and rutil. Fields Tid and iutil individually contains the identifier of Tr and the utility of X in Tr. Field rutil demonstrates the staying utility of X in Tr.

Visit Itemset Mining:

An itemset can be unmistakable as a non-void arrangement of things. An itemset with k different things is named as a k-itemset. For e.g. Consider the mix of some item 3-itemset in a market exchange. Visit itemsets are the itemsets that show up oftentimes in the correspondence. The objective of regular itemset mining is to distinguish all the itemsets in an exchange dataset. Visit itemset mining going about as a vital part in the hypothesis and routine with regards to numerous imperative information mining undertakings, like mining affiliation rules, long examples, developing examples, and reliance rules. It has been helpful in the knoll of broadcast communications, statistics investigation and content examination. The standard of being successive is articulated regarding bolster estimation of the itemsets. That estimation of an itemset is the level of exchanges that contain the itemset.

Top-k Pattern Mining:

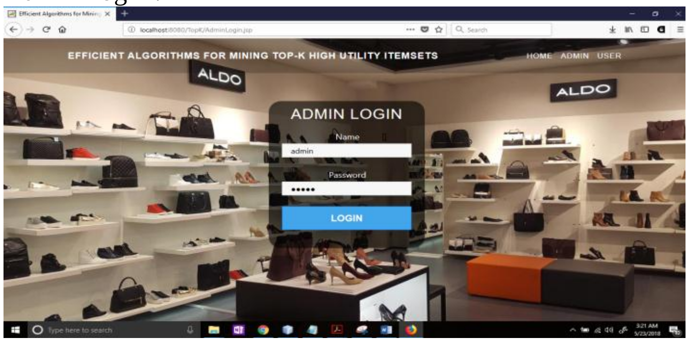
In this module, we build up the Top-k Pattern Mining. Numerous investigations have been proposed to mine various types of best k designs, for example, top-k visit itemsets top-k visit shut itemsets top-k shut consecutive examples top-k affiliation rules top-k successive guidelines top-k relationship examples and best k cosine closeness intriguing sets What recognizes each best k design mining calculation is the sort of examples found, and also the information structures and inquiry techniques that are utilized. For instance, a few calculations utilize a control extension methodology for discovering designs, while others depend on an example development seek utilizing structures, for example, FP-Tree. The selection of information structures and inquiry technique influence the productivity of a best k design mining calculation as far as both memory and execution time. In any case, the above calculations find top-k designs as indicated by conventional measures rather than the utility measure. As an outcome, they may miss designs yielding high utility

TKO (mining Top-K utility itemsets in One stage)

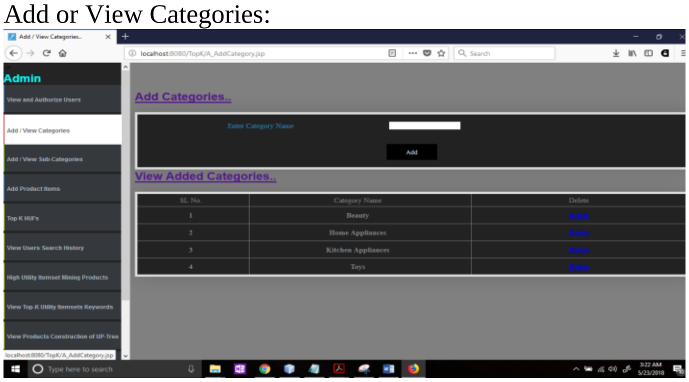
The TKO Base calculation takes as information the parameter k and a value-based database D in even configuration. In any case, if a database has just been changed into vertical configuration, for example, starting utility-records, TKOBase can straightforwardly utilize it for mining top-k HUIs. Technical knockout Base at first sets the min_util Border limit to 0 and introduces a min-stack structure TopK-CI-List for keeping up the ebb and flow top-k HUIs amid the hunt. The calculation at that point filters D twice to fabricate the underlying utility-records F-ULs. At that point, TKOBase investigates the pursuit space of best k HUI utilizing a strategy that we name TopK-HUI-Search. It is the blend of a novel methodology named RUC (Raising edge by Utility of Candidates) with the HUI-Miner seek strategy. Amid the hunt, TKOBase refreshes the rundown of current best k HUIs in TopK-CI-List and step by step raises the min_utilBorder limit by the data of TopK-CI-List. At the point when the calculation ends, the TopK-CI-List catches the total arrangement of best k HUIs in the database.

V. RESULTS

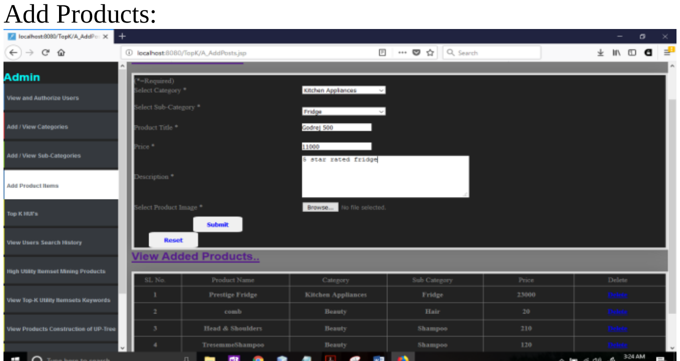
Admin Login :

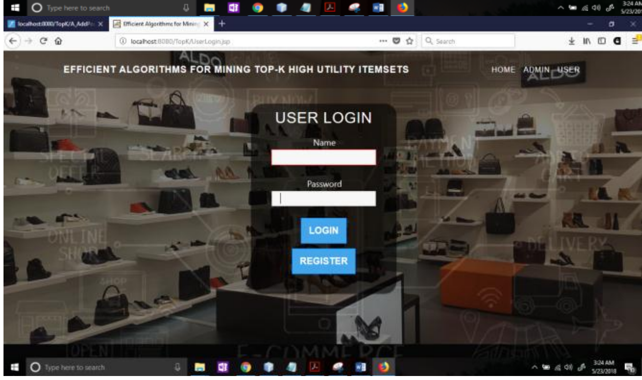


Add or View Categories:

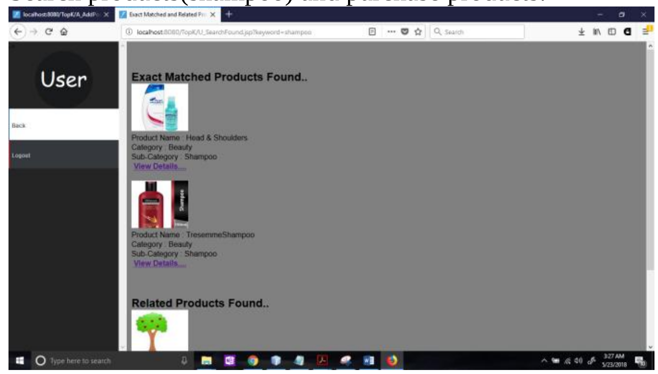


Add Products:

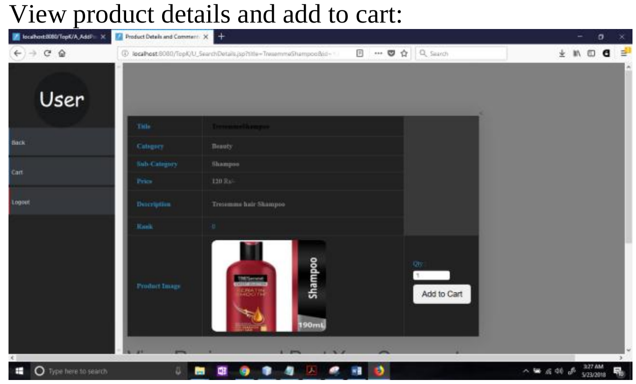




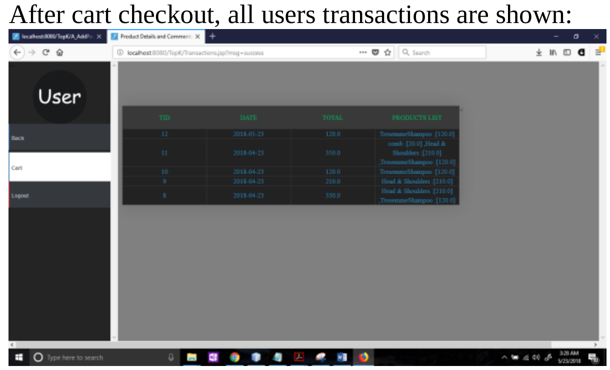
Search products(shampoo) and purchase products:



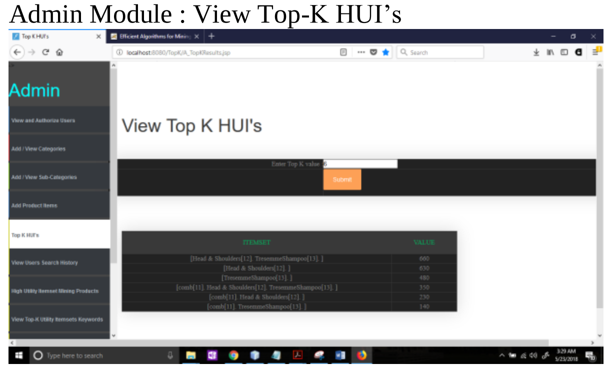
View product details and add to cart:



After cart checkout, all users transactions are shown:



Admin Module : View Top-K HUI's



VI. CONCLUSION

In this paper, we have considered the issue of best k high utility itemsets mining, where k is the coveted number of high utility itemsets to be mined. Two productive calculations TKU (mining Top-K Utility itemsets) and TKO (mining Top-K utility itemsets in One stage) are proposed for mining such itemsets without setting least utility limits. TKU is the initial two-stage calculation for mining top-k high utility itemsets, which consolidates five techniques PE, NU, MD, MC and SE to viably raise the fringe least utility edges and further prune the hunt space. Then again, TKO is the first stage calculation produced for top-k HUI mining, which incorporates the novel procedures RUC, RUZ and EPB to incredibly enhance its execution. Exact assessments on various kinds of genuine and engineered datasets demonstrate that the proposed calculations have great versatility on substantial datasets and the execution of the proposed calculations is near the ideal instance of the best in class two-stage and one-stage utility mining calculations.

In spite of the fact that we have proposed another structure for top-k HUI mining, it has not yet been joined with other utility mining errands to find diverse kinds of best k high utility examples, for example, top-k high utility scenes, top-k shut high utility item sets, top-k high utility web get to examples and best k portable high utility consecutive examples. These leave wide spaces for investigation as future work.

REFERENCES

[1] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” in Proc. Int. Conf. Very Large Data Bases, 1994, pp. 487–499.

[2] C. Ahmed, S. Tanbeer, B. Jeong, and Y. Lee, “Efficient tree structures for high-utility pattern mining in incremental databases,” IEEE Trans. Knowl. Data Eng., vol. 21, no. 12, pp. 1708–1721, Dec. 2009.

[3] K. Chuang, J. Huang, and M. Chen, “Mining top-k frequent patterns in the presence of the memory constraint,” VLDB J., vol. 17, pp. 1321–1344, 2008.

[4] R. Chan, Q. Yang, and Y. Shen, “Mining high-utility itemsets,” in Proc. IEEE Int. Conf. Data Mining, 2003, pp. 19–26.

[5] P. Fournier-Viger and V. S. Tseng, “Mining top-k sequential rules,” in Proc. Int. Conf. Adv. Data Mining Appl., 2011, pp. 180–194.

[6] P. Fournier-Viger, C.Wu, and V. S. Tseng, “Mining top-k association rules,” in Proc. Int. Conf. Can. Conf. Adv. Artif. Intell., 2012, pp. 61–73.

[7] P. Fournier-Viger, C. Wu, and V. S. Tseng, “Novel concise representations of high utility itemsets using generator patterns,” in Proc. Int. Conf. Adv. Data Mining Appl. Lecture Notes Comput. Sci., 2014, vol. 8933, pp. 30–43.

[8] J. Han, J. Pei, and Y. Yin, “Mining frequent patterns without candidate generation,” in Proc. ACM SIGMOD Int. Conf. Manag. Data, 2000, pp. 1–12.

[9] J. Han, J. Wang, Y. Lu, and P. Tzvetkov, “Mining top-k frequent closed patterns without minimum support,” in Proc. IEEE Int. Conf. Data Mining, 2002, pp. 211–218.

[10] S. Krishnamoorthy, “Pruning strategies for mining high utility itemsets,” Expert Syst. Appl., vol. 42, no. 5, pp. 2371–2381, 2015.

[11] C. Lin, T. Hong, G. Lan, J. Wong, and W. Lin, “Efficient updating of discovered high-utility itemsets for transaction deletion in dynamic databases,” Adv. Eng. Informat., vol. 29, no. 1, pp. 16–27, 2015.

[12] G. Lan, T. Hong, V. S. Tseng, and S. Wang, “Applying the maximum utility measure in high utility sequential pattern mining,” Expert Syst. Appl., vol. 41, no. 11, pp. 5071–5081, 2014.

[13] Y. Liu, W. Liao, and A. Choudhary, “A fast high utility itemsets mining algorithm,” in Proc. Utility-Based Data Mining Workshop, 2005, pp. 90–99.

[14] M. Liu and J. Qu, “Mining high utility itemsets without candidate generation,” in Proc. ACM Int. Conf. Inf. Knowl. Manag., 2012, pp. 55–64.

[15] J. Liu, K. Wang, and B. Fung, “Direct discovery of high utility itemsets without candidate generation,” in Proc. IEEE Int. Conf. Data Mining, 2012, pp. 984–989.

