Analyzing the health data: an application of high utility itemset mining

Kadium Padmavathi
Department of CSE
SRM University, Andhra Pradesh
Guntur, India
padmavathi_kadium@srmap.edu.in

Sumalatha Saleti
Department of CSE
SRM University, Andhra Pradesh
Guntur, India
sumalatha.s@srmap.edu.in

Sai Saran Tottempudi
Department of CSE
SRM University, Andhra Pradesh
Guntur, India
saisaran_tottempudi@srmap.edu.in

***Abstract*—A data science endeavour called "high utility pattern mining" entails finding important patterns based on different factors like profit, frequency, and weight. High utility itemsets are among the various patterns that have undergone thorough study. These itemsets must exceed a minimum threshold specified by the user. This is particularly useful in practical applications like retail marketing and web services, where items have diverse characteristics. High-utility itemset mining facilitates decision-making by uncovering patterns that have a significant impact. Unlike frequent itemset mining, which identifies commonly occurring itemsets, high-utility itemsets often include rare items in real-world applications. Considering the application to the medical field, data mining has been employed in various ways. In this context, the primary method involves analyzing a health dataset that spans from 2014 to 2017 in the United States. The dataset includes categories such as diseases, states, and deaths. By examining these categories and mortality rates, we can derive high-utility itemsets that reveal the causes of the most deaths. In conclusion, high-utility pattern mining is a data science activity that concentrates on spotting significant patterns based on objective standards. It has proven valuable in various fields, including the medical domain, where analyzing datasets can uncover high-utility itemsets related to mortality rates and causes of death.**

Categories and Subject Descriptors - [Health Database Application] Data Mining
General Terms - Algorithms, observations, and comparisons.
Keywords - High utility itemsets, mining algorithm, mortality rate

I. INTRODUCTION

A. Itemset Mining

To extract crucial knowledge or information, a method known as data mining is used to analyze large databases. It involves applying statistical and computer approaches to mine huge databases for patterns, correlations, anomalies, and trends. Anomaly detection, clustering, classification, regression, and association rule mining are just a few of the techniques utilized in data mining. These methods aid in revealing hidden patterns and connections between various data components. Domain experience, statistical understanding, and computational abilities are all needed for data mining [2]. The procedure entails a number of processes, such as data cleansing, preprocessing, modelling, evaluation, and deployment. Dealing with huge and complex datasets is one of data mining's challenges. Distributed computing and parallel processing are two examples of data mining technologies and methods that have been created to handle massive data. Overall, data mining is an essential technique for drawing conclusions and understanding from data to improve judgment and business outcomes [1]. The term "utility" in this context refers to a user-defined metric that depicts the usefulness, allure, or profitability of a set of things. High utility itemset mining (UH) seeks out itemsets with utility values over a minimal utility threshold, which is a user-defined cutoff. Utility in high-utility itemset mining has a similar connotation to the concept of support in traditional frequent itemset mining. Support quantifies how frequently an itemset appears in the database, whereas utility assesses the relevance or usefulness of an itemset. An itemset with a high utility value is more significant or valuable than one with a low utility value. High-utility itemset mining seeks out all itemsets that satisfy the minimal utility criterion. These costumes are said to be very practical. Candidate itemsets are formed, their utilities are assessed, and the non-compliant ones are pruned during the mining process. Once the high utility itemsets have been identified, they can be used in cross-selling, customer segmentation, and product suggestion, among other applications. For instance, a retailer might utilize high utility itemsets to identify goods that are frequently purchased together and then use this information to suggest products to customers. [3]. To find patterns, trends, and insights in huge datasets that might not be immediately obvious to the human eye, data mining uses statistical and machine learning approaches. In many different disciplines, these patterns can be used to create predictions or guide decision-making. These steps are used to turn raw data into informative material that may be applied in a variety of ways. Because it allows businesses to extract knowledge from vast amounts of data that would otherwise be difficult or impossible to examine using conventional techniques, data mining is a crucial tool for many industries. By uncovering hidden patterns and relationships in data, organizations can make more informed decisions and gain a competitive advantage in their respective markets [4]. Various data analysis techniques are employed to tackle data mining problems. Businesses utilize data mining tools to forecast future trends and make informed decisions [5]. Various optimization methods have been proposed to tackle this issue, including utility-list structures, efficient pruning procedures, and parallel computing methods [10]. The Total Weighted Utility (TWU) property is another approach employed in two-phase algorithms such as FSH, ShFSG, ZP, and ZSP [12]. TAhmed et al. developed the IHUP method, a tree-based strategy that avoids creating sizable candidate itemsets and scouring numerous databases, to address the shortcomings of earlier techniques. High Average-Utility Itemsets (HAUIs) are frequently mined using tree-based algorithms like UP-Growth and UP-Growth+. Although they only necessitate one database scan, they could be less efficient. Even though tree-based algorithms are effective at producing candidate itemsets from a single database scan, the extraction of HAUIs from these candidates may still demand a lot of work in the second phase. [12]. In the context of identifying the causes resulting in the most deaths, the HUI-miner algorithm with a utility list is employed to analyze the US deaths database. This algorithm demonstrates efficient use of space and optimal time performance, as indicated by experimental results.

B. High Utility Itemset Mining

To apply the HUI-Miner algorithm to find high-utility itemsets in the given transaction database, we can follow these steps: Step 1: Start the search procedure off with a blank itemset. Step 2: Start the depth-first search by taking into account the very first database item. Step 3: Determine whether the usefulness of the current item in each transaction exceeds or is equal to a minimal utility threshold. If so, move on to the following item in the database and add the item to the current item set. Step 4: Skip the current item in the database and go to the next one if its utility is below the minimal utility level. Step 5: Apply the same guidelines from steps 3 and 4 to the remaining database objects as you continue the depth-first search process. Step 6: Track the utility of the current itemset throughout the depth-first search and compare it to the highest utility discovered so far. Update the maximum utility value and save the current itemset as the highest utility itemset if the utility of the current itemset exceeds the maximum utility. Step 7: Return the highest utility itemset after exhausting all other possibilities.

TABLE I
DATABASE OF 5 TRANSACTIONS

T	Items	Transaction Utility	Item Utilities
t1	3 5 1	30	1 3 5
t2	6 5 2 1	20	3 3 8 6
t3	2 3 1 4	8	1 5 2 3
t4	5 2 7	27	6 6 10
t5	3 5 6 1 7	11	2 3 4 2 5

To illustrate the concept, let's consider an example using the HUI-Miner algorithm. In this example from table 1, we have a transaction database with itemsets and corresponding utility values. For instance, let's focus on transaction t1, which contains items 3, 5, and 1, with utility values of 1, 3, and 5, respectively. The sum of these utility values is 1 + 3 + 5 = 9, representing the total utility for transaction t1. For instance, if we set the minimum utility threshold to 20, the algorithm will identify itemsets that have a utility value of 20 or more. Let's take another example to better understand the utility calculation. Consider the itemset 1, 4. To calculate its utility, we need to examine the transactions where this item appears. For transaction t1, the utility values corresponding to items 1 and 4 are 5 and 6, respectively. Thus, the utility of 1, 4 in t1 is 5 + 6 = 11. Similarly, in transaction t3, the utility values for items 1 and 4 are 5 and 2, resulting in a utility of 7 for 1, 4 in t3. Adding up these values, the total utility of 1, 4 across all transactions is 11 + 7 = 18. When the HUI-Miner algorithm is used with a minimum utility threshold of 20, we discover six high-utility itemsets that meet or surpass this threshold. These itemsets may vary depending on the dataset and the minimum utility threshold chosen. They are picked based on how useful they are. Table 2, lists the itemsets, their utility values, and the corresponding support for the high-utility itemsets obtained from the HUI-Miner algorithm. Each item set's utility values across all transactions in which it appears are added up to create the utility. The number of transactions in which the itemset appears is indicated by the support. The HUI-Miner algorithm finds high-utility itemsets when they satisfy the required minimum utility threshold. According to the utility values, the resulting itemsets are chosen, and Table 2, lists the itemsets along with utility and support data gleaned from the dataset. This formula is used to determine support: Support(S) is defined as the product of the values of the column divided by the number of columns.

TABLE II
SUPPORT & ITEMSETS FOR MINIMUM UTILITY $\geq$ 20

Itemsets	Utility	Support
6, 2, 1, 5	20	20% (1 transaction)
6, 1, 5	21	40% (2 transactions)
7, 2, 5	22	20% (1 transactions)
7, 5	24	40% (2 transactions)
2, 5	23	40% (2 transactions)
1, 5	22	60% (3 transactions)

II. RELATED WORK

The unchecked proliferation of cancerous cells in the stomach lining is a hallmark of gastric cancer. Any area of the stomach can become affected, and in more severe stages, it may migrate to other organs. A high intake of smoked, salted, or pickled foods, smoking, certain genetic disorders, and a family history of stomach cancer are additional risk factors. The World Health Organization's forecast of 769,000 fatalities and one million new cases in 2020 emphasizes the significant influence of stomach cancer on world health. It is crucial to keep in mind that these figures could differ somewhat between sources and could be updated when new information becomes available. This disease has a significant impact on the health and well-being of many people around the world [14]. Making better treatment decisions is critical to increasing the survival rate of patients with gastric cancer.

These factors are critical in determining the treatment options available to gastric cancer patients. By taking all of these factors into account, healthcare professionals can choose the most effective treatment plan for each patient, increasing their chances of survival. [14]. By analyzing large amounts of healthcare data, data mining algorithms have the potential to improve healthcare delivery. The CRISP methodology, for example, provides a structured framework for data mining projects. Understanding the problem, preparing the data, modelling the data, and evaluating the results are all part of this methodology. It is possible to predict not only the mortality rate associated with gastric cancer but also the likelihood of complications following surgery, using data mining algorithms. A study was conducted to test and compare different classification models in order to improve the accuracy of predicting mortality in patients with gastric cancer. The study discovered that the J48 algorithm with oversampling was the most effective technique, with a 74% accuracy rate. This means that in 74% of cases, the J48 algorithm with oversampling correctly predicted the mortality rate of patients with gastric cancer. These findings could be used to develop more effective strategies for managing and treating gastric cancer patients, ultimately increasing their chances of survival [14]. This discovery has significant implications for improving healthcare delivery by allowing clinicians to predict and prevent potential complications in patients, ultimately improving their quality of life and outcomes [14].The prediction of mortality in patients with gastric cancer is a difficult task because it is dependent on a number of factors, including the cancer stage, the patient's overall health, and the efficacy of available treatments. However, by analyzing various health data, tumour data, and surgery information, it is possible to achieve a reasonably accurate prediction of mortality using data mining techniques. The J48 algorithm with oversampling was used in the study to predict mortality, and it had an accuracy of around 74%. Because more data is available, predicting complications after an in-hospital stay is usually easier. In the study, the Random Forest algorithm with oversampling was used to predict complications based on tumour data and the patient's health status.The study combined clustering and classification algorithms to address the high number of deaths. When the data was subdivided by gender, the model produced a matrix value of 0.495 for males and a reversible value for females. The study also divided the city into matrices, revealing that the Kasab district had the highest prevalence of malaria [15]. The study discovered that single people were more vulnerable to disease based on their marital status. Furthermore, a link between disease and gender was discovered. Males who are not married are more susceptible to malaria and have a lower risk of breast and thyroid cancer. Unmarried females, on the other hand, were found to be more susceptible to malaria and fall injuries, but less likely to have cirrhosis. These findings may be useful in identifying vulnerable populations and tailoring prevention and treatment plans accordingly [15]. The study found a link between disease and age, with unmarried people aged 5 to 47.5 being more susceptible to malaria and less susceptible to other diseases, with no cases of stroke [15]. A study was conducted using data from the UCI repository to predict heart disease in patients, using five different data mining classifiers. These classifiers are algorithms that can be trained to categorize data into various groups. Among the five classifiers tested in the study, the Bayes network classifier outperformed the others, with an accuracy rate of 79.28%. This indicates that the Bayes network classifier may be useful in predicting heart disease in patients. The Bayesian network is a statistical model that employs graphs to illustrate the relationship between variables in a dataset, allowing for a better understanding of the dependencies between factors such as smoking, diabetes, and high blood pressure, and their impact on the likelihood of developing heart disease. The study's findings suggest that these factors play a significant role in the development of heart disease. Similar to that, this paper seeks to offer fresh perspectives on what might be causing the majority of deaths via HUIM [16]. As we see there are many algorithms and methods working in health care in order to find the cause of deaths and even the algorithms used in high-utility itemset mining. Here comparing with other ideas we used a modified algorithm where will predict the set of caused which created deaths.

III. METHODOLOGY

We came up with a high-utility mining technique to find the causes which caused the most number of deaths. In this section, we will discuss the high utility mining over the health dataset of the US over a period from 2013-2017 a tenure of 5 years. This approach will lead to the high utility sets of the causes for the deaths in the US. Giving the itemsets of causes, will let us know the itemsets which are causing a high number of deaths. This helps in recognising those causes which led to the most number of deaths during 2013-2017. We used HUI Miner on the dataset fixing different amounts of minimum utilities. Table 3 is the sample data of the dataset created through resources. The algorithm is run on the US death dataset. Table 3.1 narrates countries as transactions and causes as the items. We are considering internal utility as deaths and external utility as the mortality rate of that cause in the period 2013-2017. Contemplating 10 causes by defining them as 1,2,3,4,5,6,7,8,9,10.

The causes are considered as per the analysis. Causes are renamed as 1,2,3,4,5,6,7,8,9,10. Alzheimer's disease is renamed as 1, Cancer as 2, CLRD as 3, Diabetes as 4, Heart disease as 5, Influenza and pneumonia as 6, Kidney disease as 7, Stroke as 8, Suicide as 9 and Unintentional injuries as 10. External Utility here is considered as the mortality rate. The mortality rate is calculated below

Mortality rate = Total Number of deaths /
Total Population per 1000

TABLE III
CONSIDERING 5 TRANSACTIONS FROM US HEALTH DATASET

Country	Causes	Total Utility	Utility for causes
Alabama	1 2 3 4 5 6 7 8 9 10	19718653	153300 7400965 473661 123289 10494891 74633 57848 527504 28622 383940
Alaska	2 5 10	1237925	617734 563743 56447
Delaware	2 3 5 8 10	3162326	1366540 72839 1589528 76377 57042
South Dakota	1 2 3 5 8 10	2920102	33033 1188900 75260 1474476 83131 65303
Wyoming	2 3 5 10 10	1599932	679206 56706 814528 49492

Consider the population table of 2013-2017 and calculate the 10 causes to get the mortality of the cause specifically.

TABLE IV
POPULATION OF DEATHS IN US 2013-2017[18]

Indicator	2013	2014	2015	2016	2017
Deaths	479 975	470 295	489 218	480 500	497 835
Death Rate	8.59	8.4	8.73	8.57	8.88

The total population from 2013-2017 is 280134064 from Table 4 Using this we will find the external utility for every cause. In utility mining we consider profit as the external utility here we will be considering the mortality rate of the cause as the external utility for the item and here item is the cause. The utility of the cause will be the product of external utility i.e. mortality rate and internal utility i.e. deaths. For cause 1, the mortality rate is (1471571/280134064)*1000 = 5.25, for cause 2 (10843646/280134064)*1000 = 38.71, for cause 3 (2588886/280134064)*1000 = 9.24 and similarly for other causes.

TABLE V
DEATHS AND MORTALITY RATE FOR CAUSES

Cause	Total Deaths	Mortality Rate
1	1471571	5.25
2	10843646	38.71
3	2588886	9.24
4	1363055	4.87
5	12222645	43.63
6	1061737	3.79
7	815172	2.91
8	2714571	9.69
9	663060	2.37
10	2343544	8.37

So the utility will be calculated as deaths for that cause multiplied by mortality for that cause respectively. As per table 5, we can observe that the utility of the cause is deaths and mortality rate product. In Alabama for cause 1, (29200 * 5.25) = 153300, for cause 2 (191190 * 38.71) = 7400965, for cause 3 (51262 * 9.24) = 473661, for cause 4 (25316 * 4.87) = 123289, for cause 5 (240543 * 43.63) = 10494891, for cause 6 (19692 * 3.79) = 74633, for cause 7 (19879 * 2.91) = 57848, for cause 8 (54438 * 9.69) = 527504, for cause 9 (12077 * 2.37) = 28622 and finally for cause 10 (45871 * 8.37) = 383940. Similarly for other states. When we run this data onto the HUIM algorithm with different minimum utilities we get different high utility itemsets for the deaths. So these itemsets are the set of causes which caused the most number of deaths during the period 2013-2017. The HUI-Miner algorithm is a method used to discover High Utility itemsets (HUIs) in a transactional database by setting a minimum utility threshold. The algorithm takes three inputs: a set of utility lists (ULs), a minimum utility threshold (minutil), and a Construct (P. UL, X, Y) function that merges two utility lists X and Y to create a new utility list for itemset Pxy. The algorithm follows the following steps:

ItemsetsHUIMAlgorithm1

1. The algorithm begins by iterating through a collection of utility-lists denoted as ULs.
2. For each utility-list, denoted as X, the algorithm checks if the total utility of individual items in X is greater than or equal to a specified minimum utility threshold, minutil. If this condition is met, the extension connected to X is identified as a high utility itemset.
3. To explore potential high utility itemsets further, the algorithm creates new utility-lists by merging X with each subsequent utility-list, denoted as Y, in the ULs collection. This merging process is executed if the sum of individual utilities in X and the residual utilities in X surpasses or equals minutil.
4. The resulting collection of new utility-lists generated in step 3 is stored in a variable called exULs.
5. The HUI-Miner algorithm is then recursively invoked for each of the new utility-lists in exULs. This recursive call includes the original utility-list X and the minimum utility threshold minutil as input parameters.
6. The recursion continues until all possible High Utility Itemsets (HUIs) have been generated, ensuring that no HUIs with a utility value below the specified threshold are included in the final result.

1. **Utility-Lists for Itemsets P, Px, and Py:** Utility-lists are data structures used to keep track of the utility values associated with different itemsets. In this context, three specific itemsets are mentioned: P, Px, and Py. These itemsets represent subsets of items from the dataset.
2. **Transaction-Weighted Utility:** The algorithm takes into account transaction-weighted utility, which means it assesses an item set's significance or value based on how frequently it appears in transactions. When working with transactional data, data mining frequently takes this approach.
3. **Sorting Utility-Lists:** Initially, the algorithm sorts the utility-lists in ascending order based on their utility values.

This sorted order is maintained throughout the algorithm's execution.

4. **Effective Search Space Exploration:** Utility-list sorting is essential to the effective exploration of the search space. The algorithm can efficiently intersect each utility-list Y with utility-list X in ULs (the list of utility-lists) by processing the utility-lists in sorted order. This method effectively identifies potential item sets.

5. **Minimum Utility Threshold:** The algorithm requires a user-defined minimum utility threshold. This threshold acts as a filter, ensuring that only itemsets with utility values above this threshold are considered "high utility." Itemsets with utility values below the threshold are not of interest.

6. **Recursive Processing:** The algorithm employs a recursive approach. It starts with the itemset Px and explores all possible 1-extensions of Px while maintaining the minimum utility threshold. These 1-extensions are essentially itemsets that can be created by adding one more item to Px. The algorithm repeats this process recursively, potentially generating larger itemsets.

7. **Divide-and-Conquer Strategy:** To identify all possible High Utility Itemsets (HUIs), the algorithm uses a divide-and-conquer strategy. It merges utility-lists systematically. This strategy helps manage the complexity of the mining process by breaking it down into smaller, more manageable steps.

8. **Efficiency and Effectiveness:** The HUI-Miner algorithm is efficient and effective for finding high utility itemsets in a transactional dataset because it combines recursive processing, utility-list sorting, and utility-list merging. It ensures that only itemsets meeting the minimum utility threshold are considered, reducing unnecessary computational work and improving the quality of the results.

In summary, the HUI-Miner algorithm is a sophisticated approach that leverages utility-lists, recursive processing, sorting, and merging strategies to efficiently and effectively identify itemsets with high utility in transactional datasets. It is a powerful tool for extracting valuable patterns from data where the importance of items is based on transactional context.

IV. RESULTS AND DISCUSSIONS

We considered a set of minimum utility values, specifically 1045203.339, 3135610.017, 4180813.356, 7316423.373 and so on. By running the dataset with the considered minimum utility we got a number of high utility sets, time and memory as the result. In table 6 we can see that for minimum utility 1045203.339, we got a number of high utility sets as 1023, memory as 5.430801392 MB and time taken is 22 milliseconds.

Similarly, we can see the second minimum utility i.e. 3135610.017 considered which will give the number of high utility itemsets as 1021, memory as 5.430725098 MB and time as 21ms. Homogeneously we will find the other minimum utilities as we can see in table 6 We plotted a graph considering minimum utility, utility sets, memory and time.

TABLE VI
COUNT OF HIGH UTILITY SETS, TIME AND MEMORY WITH RESPECT TO MINIMUM UTILITY

Sno	Minimum Utility	Itemsets	Memory	Time
1	1045203.339	1023	5.430801392	22
2	3135610.017	1021	5.430725098	21
3	4180813.356	1019	5.430847168	20
4	7316423.373	1016	5.430908203	19
5	8361626.712	1013	5.430755615	18
6	9406830.051	1011	5.430740356	18
7	10452033.39	1009	5.430755615	17
8	20904066.78	990	5.430725098	17
9	31356100.17	953	5.430732727	18
10	41808133.56	904	4.96875	18
11	52260166.95	865	4.96875	18
12	62712200.34	811	4.96875	17
13	73164233.73	793	4.96875	17
14	83616267.12	772	4.96875	18
15	94068300.51	768	4.96875	17
16	418081335.6	739	4.96875	17
17	522601669.5	492	4.89074707	17
18	627122003.4	256	4.890869141	19
19	940683005.1	177	4.89074707	18
20	951135038.5	141	4.89100647	16
21	961587071.9	93	4.430778503	15
22	972039105.3	58	4.430732727	14
23	982491138.7	28	3.904472351	13
24	992943172.1	14	3.904441833	11
25	1003395205	4	3.444419861	8
26	1013847239	1	3.444419861	8

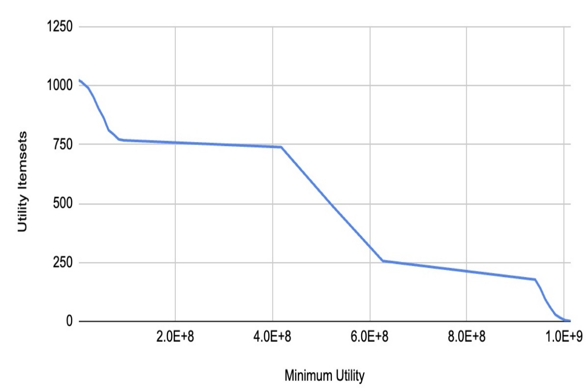

Fig. 1. Utility Itemsets vs Minimum Utility

In fig 1, we contemplated the X-axis as Minimum utility and the Y-axis as Utility itemsets count. We took a range from 0 to 8.0E+8 on the X-axis and 0 to 1250 on the Y-axis. We can see as the minimum utility increases the utility sets also increase ranging from 1023 to 1 dataset.

In fig 2, we contemplated the X-axis as the Minimum utility and the Y-axis as the Total time taken for the execution. We took a range from 0 to 8.0E+8 on the X-axis and 0 to 25 on the Y-axis. We can see as the minimum utility increased the time decreased and varied consequently ranging from 22 to 8 milliseconds.

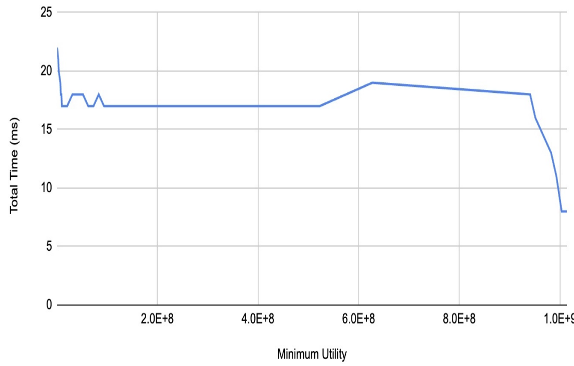

Fig. 2. Total Time vs Minimum Utility

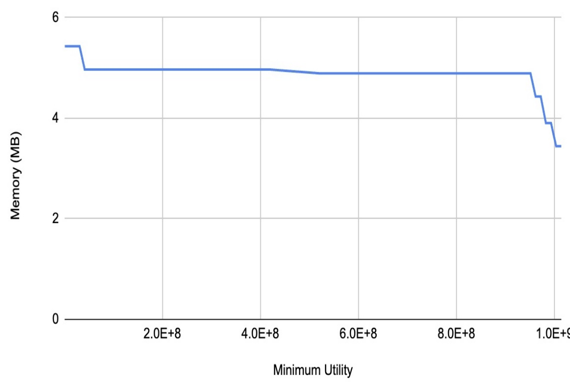

Fig. 3. Memory vs Minimum Utility

In fig 3, we contemplated the X-axis as the Minimum utility and the Y-axis as memory taken for the execution. We took a range from 0 to 8.0E+8 on the X-axis and 0 to 6 on the Y-axis. We can see as the minimum utility increased it will recast its memory changing from 5.430801392 MB to 3.444419861 MB as mentioned in the table.

V. Conclusions

High-utility itemset mining is a data mining technique used to find groups of items in a transactional database that have high utility or value. The idea of frequent itemset mining, which locates frequently occurring itemsets, is expanded by considering the utility or profitability connected to each item. It assists companies in real-time supply chain optimisation, inventory management, and product recommendations. Online advertising, fraud detection, healthcare decision support, and retail/e-commerce are some examples of applications. In this paper, we went through the US health data and came to a conclusion about the deaths. We can presume that these high utility itemsets will give the combination of causes which were causing the most number of deaths during the period 2013-2017 in the US. When we look at the result when run on min utility 1013847239 we got a high utility itemset which is [8 3 10 2 5 #UTIL: 1017985914], this 8,3,10,2,5 i.e Suicide, Diabetes, Unintentional injuries, CLRD and Influenza which a vicious combination which creates supplementary deaths. In future, we will work on the coming up years from 2017-2023 which will give the accuracy and even better prediction for cause of deaths happening and even start working on data from India.

References

[1] S. Jevalaksshmi, K. Hema Shankari, S. Mathivilasini, T.Nusrat Jabeen, K. Maheswari, D Akila, ”*Prediction and Analysis in High Utility Pattern Mining Using List Based HUPM Algorithm.*” ieeexplore.ieee.org.

[2] M. Ilayaraja, T. Meyyappan, ”*Efficient Data Mining Method to Predict the Risk of Heart Diseases Through Frequent Itemsets.*” sciencedirect.com.

[3] O. K. Alkan and P. Karagoz, ”CRoM and HuspExt: Improving efficiency of high utility sequential pattern extraction”, IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 10, pp. 2645-2657, 2015.

[4] SpringerLink : A novel way to compute association rules, 2022.

[5] J. Han, H. Cheng, D. Xin and X. Yan, ”Frequent pattern mining: current status and future directions”, Data mining and knowledge discovery, vol. 15, no. 1, pp. 55-86, 2007.

[6] M. Liu and J. Qu, ”Mining high utility itemsets without candidate generation”, Proceedings of the 21st ACM international conference on Information and knowledge management, pp. 55-64, October 2012.

[7] V. S. Tseng, B. E. Shie, C. W. Wu and S. Y. Philip, ”Efficient algorithms for mining high utility itemsets from transactional databases”, IEEE transactions on knowledge and data engineering, vol. 25, no. 8, pp. 1772-1786, 2013.

[8] R. Agrawal and R. Srikant, ”Fast algorithms for mining association rules”, Proc. 20th int. conf. very large data bases VLDB, vol. 1215, pp. 487-499, September 1994.

[9] A. Inokuchi, T. Washio and H. Motoda, ”An apriori-based algorithm for mining frequent substructures from graph data”, European Conference on Principles of Data Mining and Knowledge Discovery, pp. 13-23, 2000, September.

[10] J. Han, J. Pei, Y. Yin and R. Mao, ”Mining frequent patterns without candidate generation: A frequent pattern tree approach*”, Data Mining and Knowledge Discovery, vol. 8, no. 1, pp. 53-87, 2004.

[11] Y.-C. Li, J.-S. Yeh and C.-C. Chang, ”Direct candidates generation: A novel algorithm for discovering complete share-frequent itemsets”, Proc. Fuzzy Systems and Knowledge Discovery, pp. 551-560, 2005.

[12] V. S. Tseng, C.-W. Wu, B.-E. Shie and P. S. Yu, ”Upgrowth: An efficient algorithm for high utility itemset mining”, Proc. ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining, pp. 253-262, 2010.

[13] Y.-C. Li, J.-S. Yeh and C.-C. Chang, ”A fast algorithm for mining share-frequent itemsets”, Proc. AsiaPacific Web Conf., pp. 417-428, 2005.

[14] Cristiana Neto, Maria Brito, Vítor Lopes, Hugo Peixoto, António Abelha, José Machado, ”*Application of Data Mining for the Prediction of Mortality and Occurrence of Complications for Gastric Cancer Patients.*” mdpi.com.

[15] Ahmed Salah al-Din Abdullah, “USING DATA MINING TECHNIQUES TO IDENTIFY THE CAUSES OF DEATHS IN ALGEDAREF HOSPITAL”, European Journal of Computer Science and Information Technology Vol.4, No.2, pp.1-8, March 2016.

[16] Prasenjit Dasa , Shaily Jainb , Chetan Sharmac , Shankar Shambhua , Sakshid, “Prediction of Heart Disease Mortality Rate Using Data Mining”, pp. 352-367, Inderscience Publications, 2015.

[17] https://www.kaggle.com/datasets/cdc/nchs-death-rates-and-causes-of-death?resource=download&select=nchs-leading-causes-of-death-united-states.csv.

[18] https://public.knoema.com/lbhargc/national-population-by-state-u-s.