Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 218 (2023) 174–188

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Machine Learning and Data Engineering

Machine Learning based Approximate Query Processing for Women Health Analytics

A J Parvathi[a], Gopika H[a], Jasna Suresh[a], Sapa Laasya Sree[a], Sandhya Harikumar[a]

[a]*Department of Computer Science and Engineering, Amrita Vishwa Vidyapeetham, Amritapuri, India*

Abstract

Good health and well being is one of the most essential targets of the Sustainable Development Goals (SDGs). This paper primarily focuses on Preventive and Diagnostic care of Women Health because even today, women are disadvantaged by discrimination in many societies especially in rural sectors. Two main health issues, fetal abnormality in pregnant women and cervical cancer in women are analyzed so that the doctors and patients can be given early signals to take proactive measures. As per National Health Portal of India, around 1.7 million birth defects occur in India every year. So Antenatal Care(ANC) should be given utmost importance during Pregnancy in a woman's life. Cervical cancer is another issue prevalent amongst women, especialy over the age of 30. It's critical to catch it early and eliminate any risks that come with it. Here arises the need to develop a system that can analyze and predict the aforementioned anomalies at an early stage. This work proposes approximate query processing using different dynamic machine learning algorithms to analyze and predict the abnormalities. Further, a web application is built to facilitate the stakeholders, especially doctors, to interact with the system and iteratively query the system to understand the relationships amongst the various data variables and get appropriate predictions about fetal anomaly from CTG scans as well as presence of cervical cancer from various demographic information, habits, and historic medical records. Approximate Query Processing is accomplished using Correlation analysis, linear, and logistic regression algorithms in a dynamic, interactive, and iterative manner.

Keywords: CTG; Approximate query processing; fetal anomaly; cervical cancer

1. Introduction

Sustainable Development Goals(SDGs) are put forth to end various types of deprivations, by developing strategies for the goodness of health and education, to reduce inequality prevalent in various societies. For example, the SDG 3.8 target aims to "achieve Universal Health Coverage(UHC), including financial risk protection, access to quality

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000.
E-mail address: author@institute.xxx

1877-0509
10.1016/j.procs.2022.12.413

essential health care services, and access to safe, effective, quality, and affordable essential medicines and vaccines for all." [1]. There are societies existing, where women are health illiterate and are not given enough consideration especially in rural areas. Our work deals with analytics of two severe issues concerning women, pregnancy and cervical cancer. According to the WHO, 303,000 women died from pregnancy-related reasons in 2016, 2.7 million babies died within the first 28 days of their lives, while 2.6 million babies were stillborn.[2]. Many of these fatalities might be avoided with proper fetal and neonatal care, yet only 64 percent of women worldwide receive antenatal care four or more times during their pregnancy. During pregnancy, the fetal period begins 8 weeks after fertilization and ends at the time of child birth. It is very critical to monitor the fetus periodically to ensure that it is healthy. It is critical for pregnant women to get frequent Antenatal Care during each trimester to ensure the baby's health and well-being[1]. Pregnant women must also be conscious of their fetal health as well as their own health status, which necessitates regular exams and examinations of the fetus, as well as being informed about the right food plans and nutrient intake that they should follow. The cardiotocography (CTG) scan is used to monitor the mother's uterine contractions as well as scan for any abnormalities in the fetus[2]. When an obstetrician-gynecologist studies the CTG scan result, they will be able to gather all of these details and advise the pregnant lady about their fetus's health condition, allowing them to take the required actions to ensure the fetal health is normal.

Cervical cancer, which develops in the cervix, or uterine neck, is another serious problem. It is one of the most common causes of death in women, according to [3, 4]. It's possible that the cancer will spread to other parts of the body if it develops beyond the cervix. All women, especially those between the ages of 30 and 49, should undergo a gynaecological examination and test to identify if they have a precancerous lesion in their cervix or are at high risk of developing one as a result of an HPV infection. Cervical cancer is one of the most effectively treatable cancers once detected, as long as it is caught early and treated appropriately. As a result, early detection and testing can considerably lower any form of risk. [3] So, a reliable approach for detecting cervical cancer in women at an early stage is necessary. Patients data is usually extremely extensive and frequently updated at hospitals and medical institutions. To analyze massive data and explore the intricacies in such a dynamic and real-time environment is not only time consuming but also demand for flexibility in the querying system. So there is an inevitable need for an efficient system to process and examine the data. However, present models require a long time to process such large amounts of data. The proposed solution is approximate query processing based on various types of machine learning algorithms that generates model as per the user query to predict fetal anomaly and detect cervical cancer. Approximate Query Processing not only aids in lowering latency but also generates the results in a much more efficient and timely manner.

A web application is built that provides a platform to answer mainly three types of queries. One is to understand the correlation between the various features of the dataset. The second one is to check how much a subset of features influence some other feature. The third one is to predict the target class based on the entire or subset of features. The user has the provision to choose one of the databases stored as per the requirement. In this work, one is CTG scan results for fetal abnormality and the other one is cervical cancer. The users of this system are doctors and health experts.

In order to process these queries, approximate query processing has been used so that the system exhibits high-level efficiency at a much faster rate. Approximate Query Processing is a technique that offers an approximate answer based on information that is comparable to the information used to answer the question with a shorter response time[5, 6]. In this work, rather than executing the same queries in a repetitive manner, approximate query processing stores the results of the previous queries and thus enables the database to be faster at answering new queries since the whole data in the database is not being searched every time.

Using Approximate Query Processing, the aim is to build a model from the queries and the answers corresponding to those queries. The advantage of building such a model is that, when a query is executed, instead of processing the output from raw data, it will retrieve the information from previously executed queries to generate the output of the current query. This model allows the database to be faster at answering new queries since the whole data in the database is not being searched every time when a query is executed, thereby reducing its response time.

[1] https://www.worldbank.org/en/topic/universalhealthcoverage

[2] https://www.who.int/news/item/07-11-2016-pregnant-women-must-be-able-to-access-the-right-care-at-the-right-time-says-who/

[3] https://www.ncbi.nlm.nih.gov/books/NBK269601/

The main contributions of this work are:

- Approximate Query Processing using correlation analysis, dynamic linear regression and logistic regression on real-time data.
- A time efficient intelligent query system in the form of web application to facilitate interactive and iterative querying for women health analytics.

The paper is organized as follows. Section 2 discusses the literature survey that elaborates various other solutions in similar context. Section 3 elucidates the proposed methodology, including the queries framed for analytics, the block diagram of the solution and the corresponding algorithms. Section 4 demonstrates the experimentation and comparison of the traditional methods with the solution proposed. Section 5 illustrates the results obtained as a part of the experimentation, and Section 6 is on the conclusion and plausible future work.

2. Literature Review

Various techniques for Approximate Query Processing has been attempted. Most of them are statistical based and a few of them have machine learning aspects. But those lack analytical query processing. There are possibilities of using generative models for approximate query processing as discussed in [5]. There are two techniques for query processing that have been discussed. These are namely probability and expectation estimates-based and another one that is based on a technique called sampling. It also analyzes using a detailed experimental evaluation based on actual and simulated data sets to compare alternative query processing algorithms. Two different strategies mentioned include Probability-based and Sample-based. The proposed techniques' are a good choice for analyzing huge datasets. But the major limitation for these techniques is that they do not support complex queries.
Various online and offline AQP methods are illustrated in [6]. Here, the issue of extending approximate query processing to new applications and to process complex data are presented. It also provides a detailed description about the various challenges faced while using existing AQP techniques: online aggregation and offline synopses generation. Other complicated data, such as geographical and trajectory data, can also benefit from AQP approaches. It can also be used to improve data visualization and data cleaning in other applications. But the approach mentioned lacks a well formulated AQP framework and calls for designing new algorithms to deal with large datasets with non-Gaussian distribution data. An idea called Database Learning(DBL), which produces results for future queries by learning from the past queries is presented in [7]. It leverages estimated replies to previous searches as observations to improve our understanding of the underlying data, which can then be utilised to speed up future inquiries. On top of Spark SQL, a query engine named Verdict is created, and experiments are run. Verdict gives up to a 23-fold speedup and a 90 percent error reduction when compared to AQP engines that do not use DBL. DBL translates a wide range of SQL queries into mathematical representations that may be put into statistical models and used to enhance query accuracy in the future. In [8], the pros and downsides of existing error estimating approaches are discussed.It even explains how to put up a pipeline for the approximation of queries that provides accurate results and error bars at dynamic speeds. It delves into the architecture of the error estimation pipeline. However, this method may not always ensure efficient and precise results. Answering questions based on data synopses in [9] is a cost-effective way to handle vast amounts of data. It takes two approaches: a pre-computed synopsis and query processing in real time. The two key characteristics described are data reduction and aggregate queries, both of which are important for successful AQP results and are complementary. Histograms have a use restriction since they require significant changes to query processors and query optimizers to use them for online query processing. Wavelets, like histograms, have the drawback of requiring changes to query processors and query optimizers to use. They're now restricted to a few types of aggregate queries and aren't very versatile. A dynamic sample selection architecture for producing rapid, approximate answers to analytical queries across huge databases is discussed in [10]. According to the paper, selecting relevant bits of previously produced samples can offer more accurate approximate answers than using uniform or non-uniform samples in a static, non-adaptive manner. However, dynamic sample selection is not appropriate for all AQP applications. Comparison of AQP to query-time sampling is depicted in [11]. AQP techniques based on pre-computed samples yield a significant reduction in response time, although they are confined to a subclass of aggregate queries. According to research and findings from various research works, approximate query processing is a growing data processing technique. Query

response time can be greatly lowered, which can be useful when dealing with extremely huge amounts of data. As a result, this technology has the potential to be immensely valuable in fields such as healthcare, astronomy, finance, and so on. Adapting approximate query processing techniques and tailoring it based on the use case at hand can assist increase the efficiency of the results produced.

3. Proposed Methodology

Data is continually expanding at a breakneck pace. Processing massive amounts of data and extracting the appropriate results is a tremendous undertaking. So we propose machine learning based Approximate Query Processing[12] that can be utilized to streamline this procedure while maintaining the accuracy of the query results. Using AQP, the results of a query execution are saved and later used when the query is conducted again at some point in the future. As a result, AQP aids in lowering latency and generating results in a much more efficient and timely manner.

3.1. Dynamic Machine Learning Algorithms

When the data is stored within DBMS, the users have the provision to query the subset of features and subsets of samples. Conventional Machine learning algorithms work on static historical data[13] to learn from data and generate model to predict the target feature. But querying cannot be restricted to predicting just the target feature and that too from historical data. So we propose Dynamic Machine Learning algorithms that analyze the data for predicting the required feature based on subsets of features in real-time environment. Our system provides three analytics query that can execute on subset of features and samples in real time to predict any other required feature as well as explore the correlation amongst them. This leads to a variant of Approximate Query Processing based on Dynamic Machine Learning. Currently, the queries proposed are as follows, but this is a generic framework where more queries can be embedded such as dimensionality reduction, matrix factorization[14], etc.

- Prediction Query : Predict the value of a feature value based on the given subset of feature values in the data. We do not restrict the user to predict only the original target feature.
- Correlation Query : Find the correlation of a feature based on the given subset of features. Pearson correlation coefficient is used to do the correlation analysis. Such analysis not only help the user to understand how are the features correlated but can also help him to identify the importance of different features and how it contributes to prediction of the abnormalities.
- Classification Query: Given a new patient record pertaining to pregnancy or cervical issues, classify her corresponding health status. Here its not mandatory to input all the features. We adapt ML algorithms such as logistic regression to dynamically generate the model for predicting the target feature based on the subset of features given by the user.

A web application is deployed that uses dynamic machine learning algorithms for approximate query processing and aids in the early detection of fetal anomalies and detection of cervical cancer. End users are presented with a user-friendly interface to interact with the data and get instant results using Approximate query processing via a web application. It also makes it more accessible and available, and it eliminates the need to master new technologies in order to use it. Using the online application, they acquire the required results sooner and in a more easily understandable format. User can choose the type of database and the type of queries they want to execute, on any of the datasets they have stored in the database. The block diagram of the system is shown in Figure 1
In this work, we primarily focus on women's health issues such as fetal health during pregnancy and cervical cancer[15]. Different machine learning models are built dynamically based on the query given by the user. The features in the query are considered to generate models using machine learning algorithms such as linear regression and logistic regression as shown in the block diagram in Fig. 1. Thus, in contrast to traditional machine learning algorithms, a system for implementing Approximate Query Processing using dynamic machine learning algorithms has been developed. The queries may be dynamically entered based on the stakeholder's needs in terms of different subsets of features. The system generates the model dynamically with different subset of features and stores the various models rather than directly generating it each time when the queries are input by the user. Thus, whenever a new query

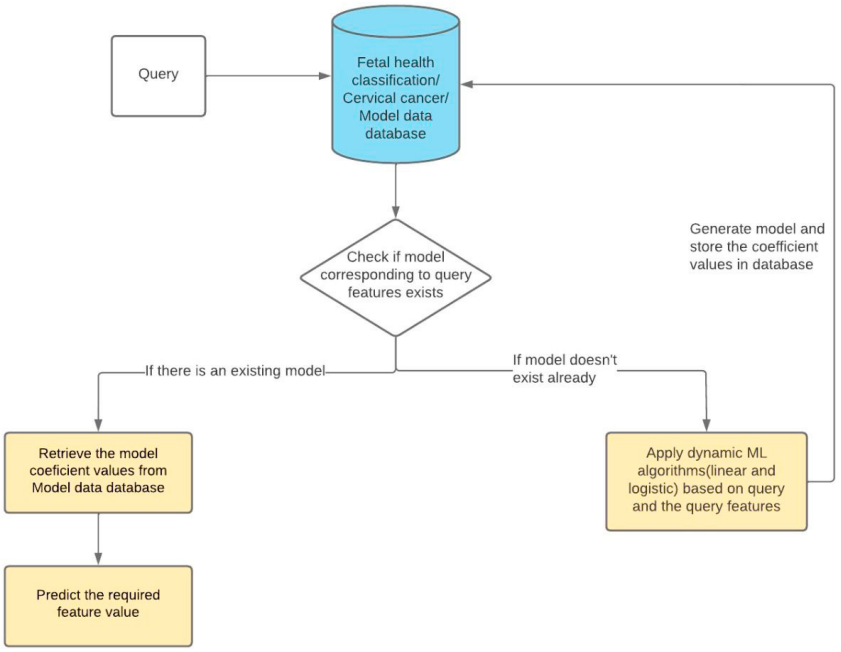

Fig. 1: Block Diagram

is executed, it first checks the database to see if the query has already been executed. If the result of a new query is found to be pre-existing in the database as it has been executed earlier, the apt model is retrieved from the database and the result for the current query is computed rather than generating a new model. Thus dynamic linear regression and logistic regression models are generated to perform approximate query processing. In contrast to the conventional machine learning algorithms, the results are obtained at a much faster rate using this approach.

3.2. Algorithms

Existing model: When a new query is run, the database is verified if it has been run previously. If the currently executed query's model are already in the database, the algorithm uses the existing model to calculate the target feature value.

Algorithm 1 Pseudocode for existing model

1: **Input** Subset of Features with corresponding values; Required Target feature name
2: **Output** Target feature value
3: Retrieve the coefficient values from the database for the given input subset of features
4: **for** each record in the features' coefficient values **do**
5: Apply linear regression
6: $h_\theta(x) \leftarrow \theta^T.X$, where θ and X are column matrices
7: **end for**

Generate model: When a query is run, the database is checked if it has been executed previously. If the query is found to be executed for the first time, i.e the coefficient values are not present in the database, generate model is used. The function does linear/logistic regression on the given set of features and their corresponding values to get the target feature value and stores the obtained coefficient values to the database.

Algorithm 2 Pseudocode for Generate model

1: **Input** Features and corresponding values
2: **Output** Target feature value
3: Perform linear/logistic regression
4: Store the output(coefficient values) in an array
5: **for** each record in the features' coefficient values **do**
6: Apply linear/logistic regression
7: $h_\theta(x) \leftarrow f(\theta^T.X)$, where θ and X are column matrices and f is a linear/sigmoid function in linear/logistic regression respectively
8: **end for**
9: Return target feature value based on the range to which $h_\theta(x)$ belongs to

The fetal health csv and cervical cancer csv file is imported into the PostgreSQL database, and the linear regression model is then performed to calculate the target feature value. The data and results of the executed queries are stored in the database. The features and their corresponding coefficient values that are obtained by the Generate model function are also stored in the database but in a separate table called "Modeldata". This table is used by the Existing model function to check if the query is executed previously and use the corresponding values. The table "Modeldata" has two columns - first with the set of features and the second column contains the coefficient values corresponding to the model generated during the particular query.The fetal health classification table has 2126 records with 22 features each. The cervical cancer table has 72 fields with 20 attributes each and the cervical cancer risk detection table has 858 records with 36 features each.

4. Experimentation

4.1. Setup

The implementation and experimentation is carried out and tested on a commodity machine with an $i3$ $7th$ generation processor clocked at $2.29GHz$ with $4GB$ RAM. For the experimentation, three datasets have been used, a fetal health dataset showing different CTG(cardiotocogram) results, a cervical cancer detection dataset and a cervical cancer risk classification dataset are used. The first dataset is the report of different cardiotocogram reports and the target feature of this dataset is fetal health which takes three values 1,2 and 3 which classifies fetal health into three classes. If the fetal health value is 1 then its classified as normal, if its value is 2 then its classified as suspect and fetal health is classified as pathological if its value is 3. The second dataset tells whether cervical cancer is detected or not. Here, in this dataset the target feature takes two values 0 and 1. If the target feature value is 1 then cervical cancer is detected and if it takes the value 0 then cervical cancer is not detected. Third dataset used is a cervical cancer risk classification dataset which is the report of different biopsy reports. Here, the target feature is biopsy which takes two values 0 and 1. If the biopsy value is 1 then cervical cancer is detected, else it is not detected.
The web application is created on a personal computer and can be used with any simplified code editor, such as Visual Studio Code. HTML, CSS, and JavaScript are used to implement all of the front-end features. The framework utilised is Flask, and all of the key functionality have been coded in Python.
The system uses a PostgreSQL13 with pgAdmin4 database to store the dataset and the coefficient values of the executed queries in order to employ approximate query processing using dynamic machine learning models. For each dataset used, three tables are used: one for storing the entire dataset result, another for storing the set of features queried by the user and their corresponding coefficient values generated using the dynamic linear regression model, and a third for storing the set of featured queried and coefficient values generated using the dynamic logistic regression model. The second and third tables help in performing approximate query processing, as these are queried first while executing a new query to check if a model has already been generated for the current set of inputs. If this is not the case, a model is constructed and a new entry is added to the associated table.

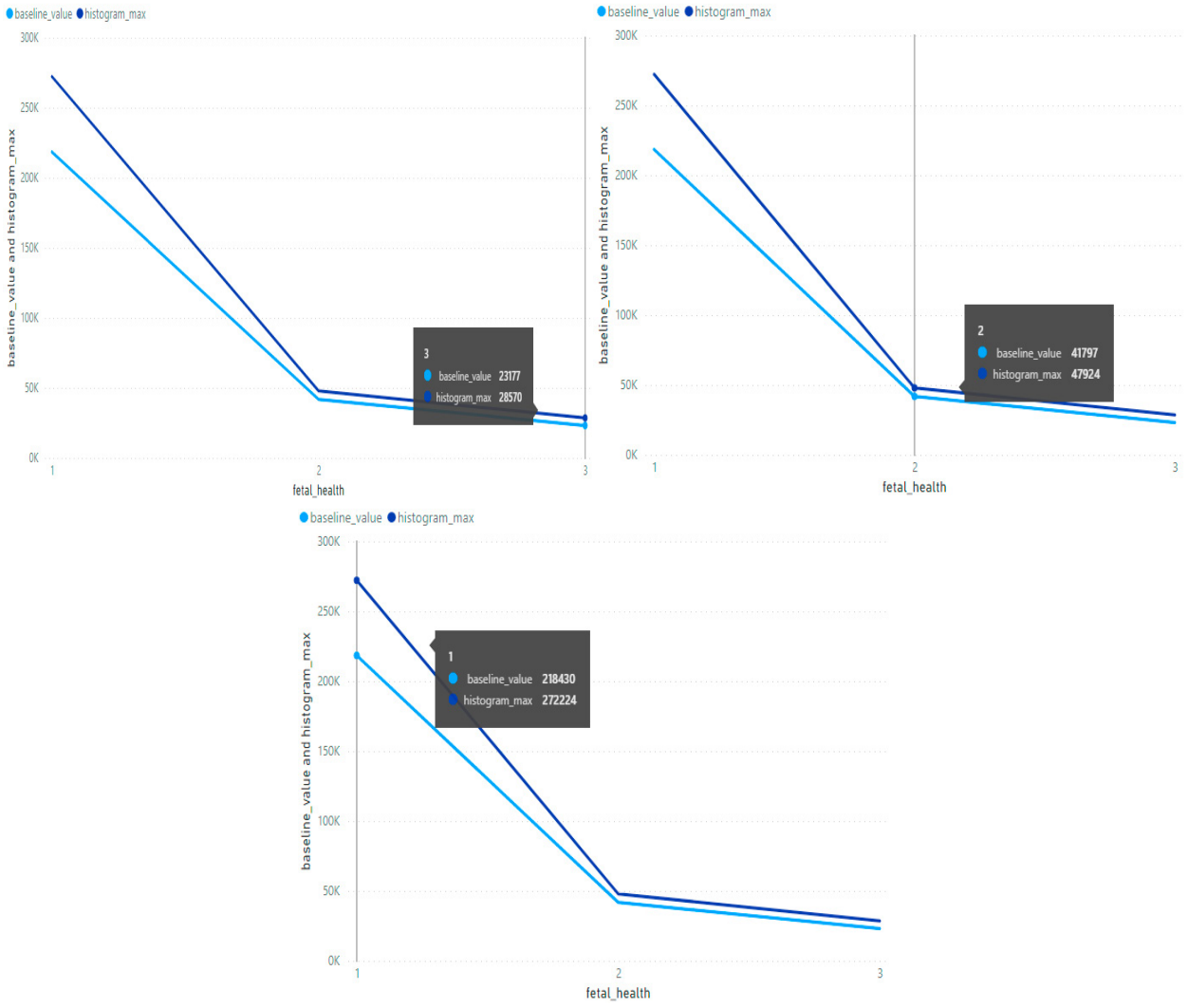

Fig. 2: Correlation of different set of features with respect to fetal health

4.2. Comparison of the developed system with VerdictDB

VerdictDB is a query accelerator[16] that allows the application to execute queries to receive an approximate answer immediately. As a result, VerdictDB can also be used for approximate query processing. However, the results showed that VerdictDB is more suited for use when working with aggregate functions and scanning large amounts of data. In such circumstances, the query's execution time is shortened. In the proposed application, however, the execution time for queries executed through VerdictDB proved to be longer than the time required by the dynamic machine learning models that were developed. Fetal health and cervical cancer classification is a sensitive use case that necessitates reliable answers to the queries posed. As a result, VerdictDB is ineffective. It also necessitates the installation of a Java library between the application and the database, resulting in a more complex application architecture.

4.3. Execution Time

The execution time for a new query that isn't in the existing model database, i.e. for a query whose model hasn't been created yet, was calculated, as well as the execution time for a query whose model has already been created, i.e.

for a query whose data has already been stored in the existing model database. It was discovered that the latter took less time to execute because it didn't have to process the entire database and generate a new model to retrieve the query's response. Fig.3 shows a plot between the execution time of query processing based on the model generated using generate model function versus the time taken when the system uses an already existing model generated prior. The graph clearly proves the efficiency of the system that has been developed in terms of reducing execution time effectively with the help of approximate query processing and dynamic machine learning models.

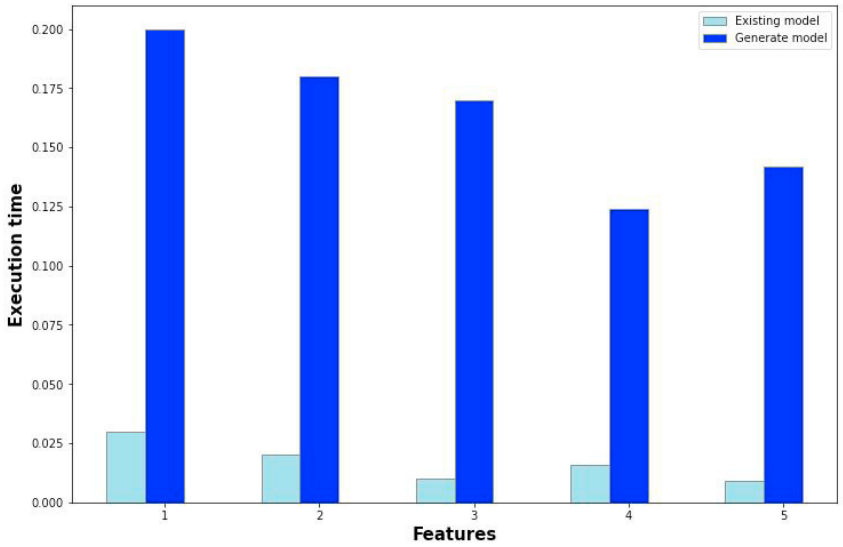

Fig. 3: Comparison between execution times of query processing using existing models and generate models

4.4. Performance metrics

Table 1 shows the performance metrics for the dynamic logistic regression model for fetal health dataset.

Table 1: Performance metrics for cervical cancer dataset.

Accuracy	84
Precision	
Class 1	0.89
Class 2	0.57
Class 3	0.74
F1 Score	
Class 1	0.91
Class 2	0.48
Class 3	0.75

Table 2 shows the performance metrics for the dynamic logistic regression model of cervical cancer dataset.

Table 2: Performance metrics for cervical cancer dataset.

Accuracy	93
Precision	
Class 0	0.96
Class 1	1.00
F1 Score	
Class 0	0.98
Class 1	0.91

Table 3 shows the performance metrics for the dynamic logistic regression model of cervical cancer risk classification dataset.

Table 3: Performance metrics for cervical cancer risk classification dataset.

Accuracy	90
Precision	
Class 0	0.91
Class 1	0.57
F1 Score	
Class 0	0.95
Class 1	0.24

Rmse(Root mean square error) values[17] are calculated and plotted for all the three datasets as shown in Fig.4, Fig.5 and Fig.6.

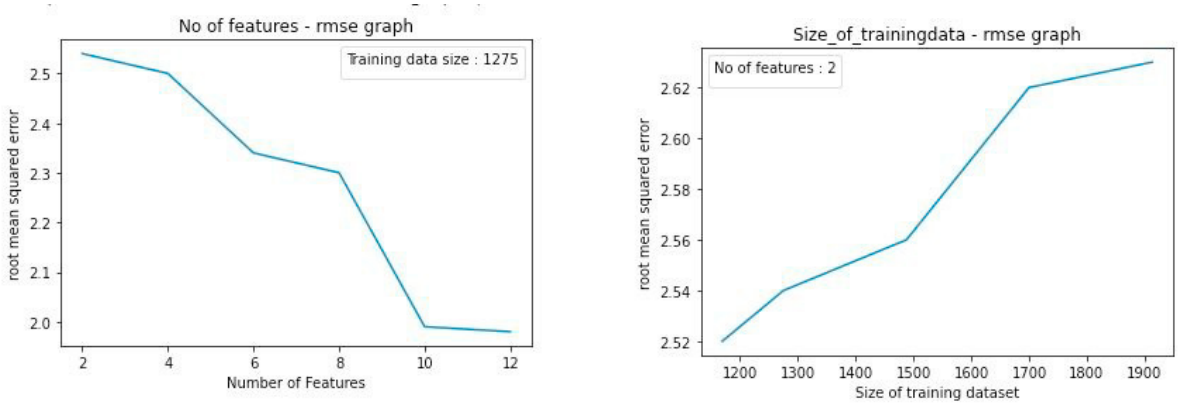

Fig. 4: Rmse graph wrt increase in number of features and wrt increase in size of training dataset of fetal health classification dataset.

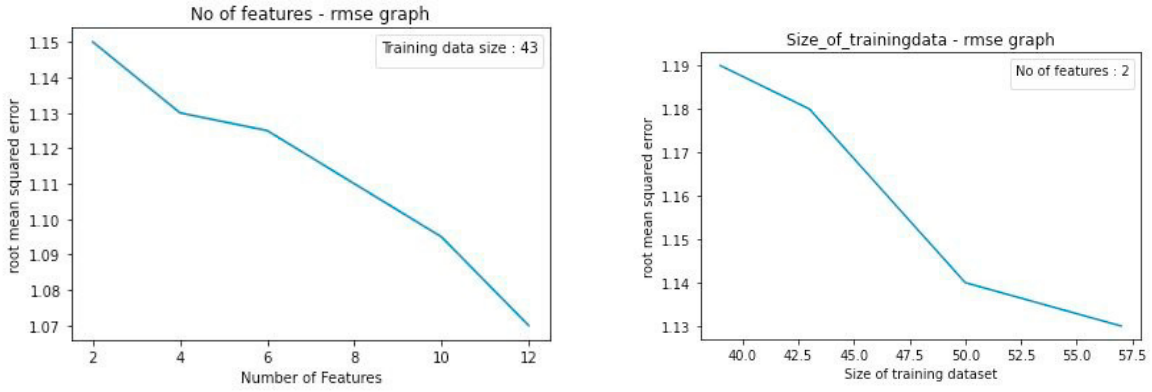

Fig. 5: Rmse graph wrt increase in number of features and wrt increase in size of training dataset of cervical cancer dataset.

Two Rmse plots are shown for each of the dataset, one shows the rmse values wrt increase in number of features and is observed that rmse value decreases when the number of features increases and the second plot shows the rmse value wrt size of training dataset and is observed that rmse value decreases when size of the training dataset increases.

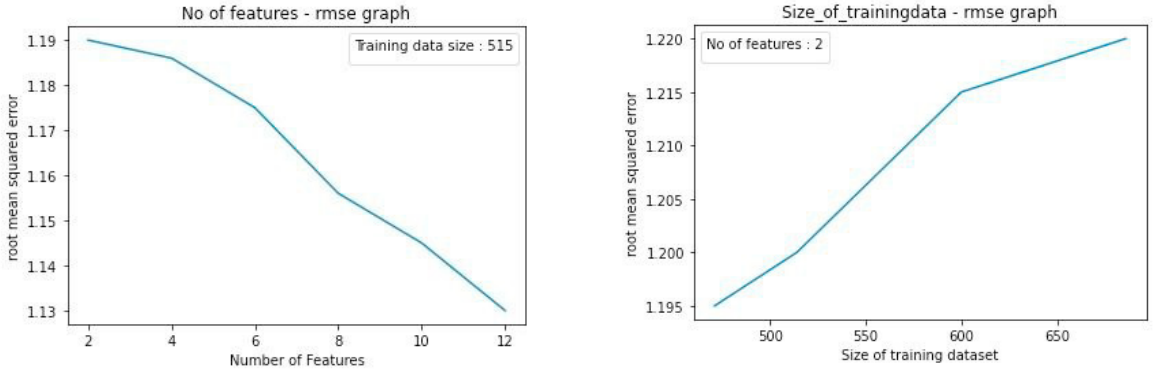

Fig. 6: Rmse graph wrt increase in number of features and wrt increase in size of training dataset of cervical cancer risk classification dataset.

Correlation between different features in each of the dataset are calculated using pearson coefficient method. Fig.7 shows positive, negative and zero correlation graphs for the fetal health classification dataset. Graph between baseline_value and histogram_mode shows the positive correlation, graph between histogram_width and histogram_min shows negative correlation and graph between severe_decelerations and uterine_contractions shows zero correlation.

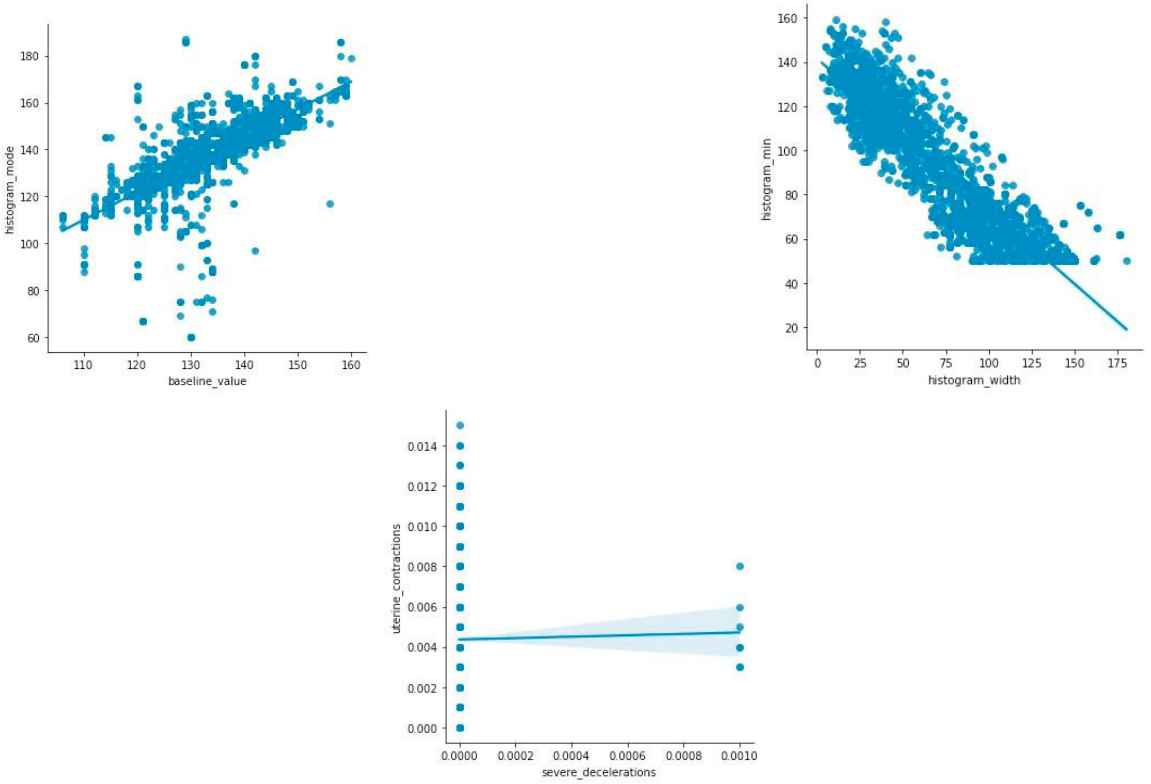

Fig. 7: Graph showing positive, negative and zero correlation of fetal health classification dataset

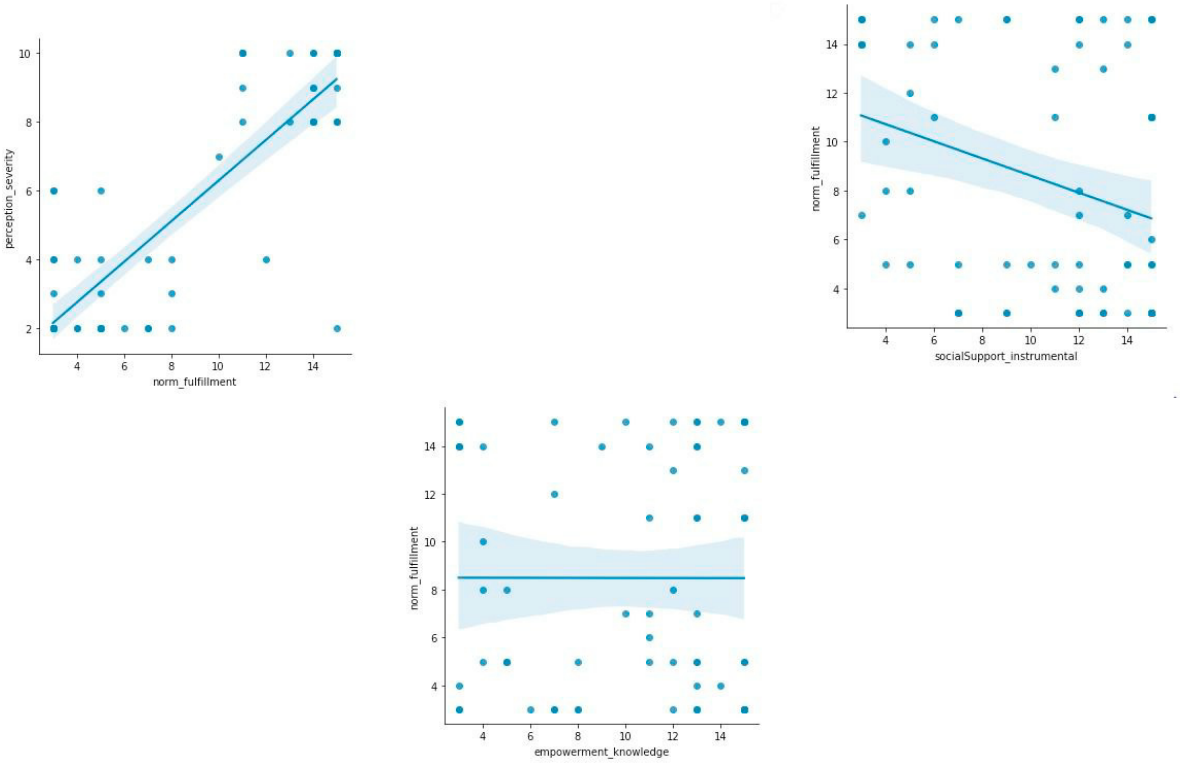

Fig. 8: Graph showing positive, negative and zero correlation of cervical cancer dataset

Fig.8 shows positive, negative and zero corelation graphs for the cervical cancer dataset. The plot between norm _fulfilment and perception _severity shows positive corelation, the graph between socialSupport _instrumental and norm _fulfilment shows negative corelation and the plot between empowerment _knowledge and norm _fulfilment shows zero corelation.

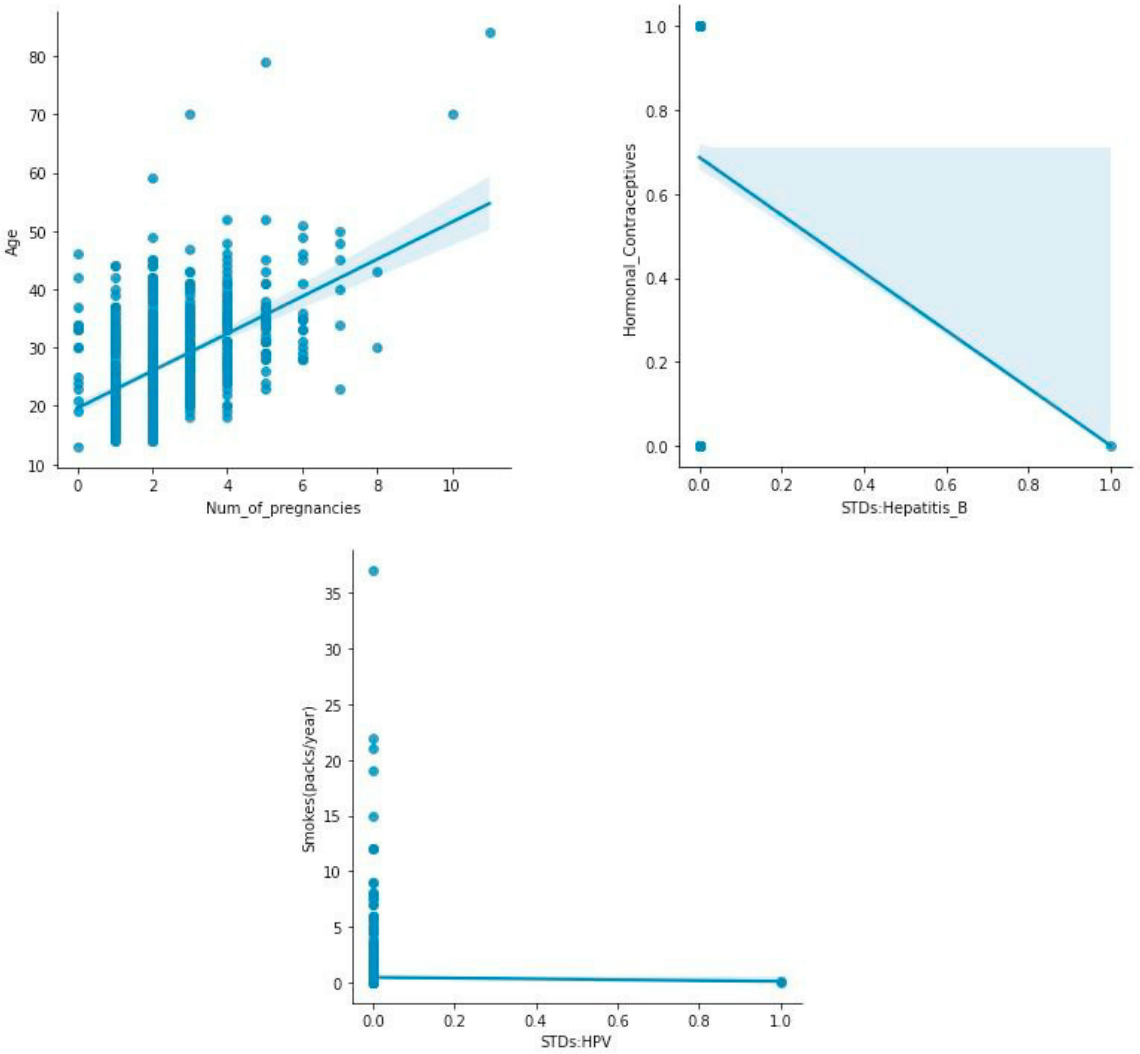

Fig. 9: Graph showing positive, negative and zero correlation of cervical cancer risk classification dataset.

Fig.9 shows positive, negative and zero correlation graphs for the cervical cancer risk classification dataset. Graph between Age and Num_of_pregnancies shows the positive correlation, graph between Hormonal_Contraceptives and STDs:Hepatitis_B shows negative correlation and graph between Smokes(packs/year) and STDs:HPV shows zero correlation.

5. Results

With the philosophies and methods discussed in the previous parts, the website was successfully built. The website is currently dynamically operational, with a front-end driven by HTML, CSS, and JavaScript, and it meets its core goal of employing machine learning to answer the questions described in the preceding section. It was developed in Python

using the flask framework. The homepage of the application is shown in fig. 10.The homepage of the application allows users to navigate to different sections of the website.

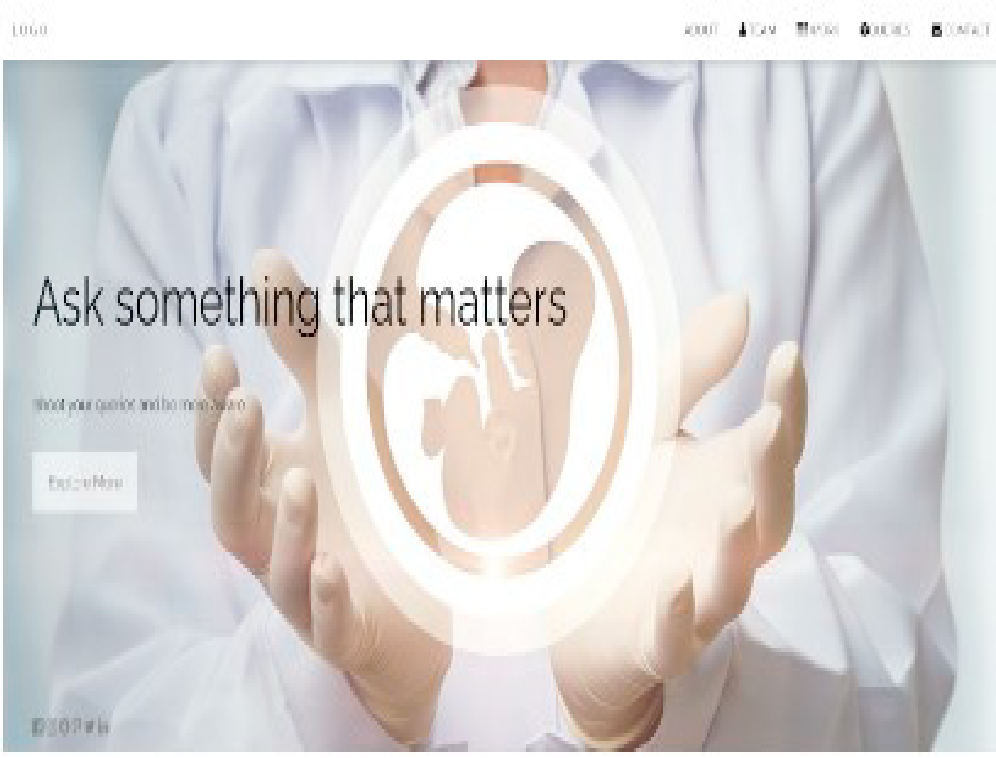

Fig. 10: Homepage of the application

The first query takes features and their corresponding values and a target feature as inputs and returns the value of the target feature with respect to the entered features and its values. Fig. 11 shows the result of the first query in the fetal health section.

Fig. 11: Result of first query on fetal health classification dataset

Fig. 12 shows the correlation between features given as inputs in the fetal health section. This query can be used to find out the dependence of a feature on various other features as well as determine the relation between different sets of features.

Fig. 12: Result of second query on fetal health classification dataset

The third query in the fetal health section displays the status of fetal health - as normal, suspect or pathological based on the features and their corresponding values given as inputs. The third query in the cervical cancer section shows if cervical cancer is detected or not. Fig. 13 shows the result of third query in fetal health section.

Fig. 13: Result of third query on fetal health classification dataset

6. Conclusion and Future Work

This application uses the concept of approximate query processing using which the information from previously executed queries is retrieved in order to generate the output for the current query and this makes the model run faster and improves the overall efficiency of the solution in contrast to the existing systems that use traditional machine learning algorithms to solve the problem. Based on our experimentation, it has been proven that using approximate query processing, the efficiency of the system can be improved by many folds when compared to the traditional approaches. This application can bring a huge transformation in the field of healthcare. It can be used to estimate fetal health and informs the user if there are any fetal abnormalities at an early stage of their pregnancy and the application can also be used to detect cervical cancer so that they come to know about it at an early stage which makes the

treatment easier and thus makes it easily curable . The system can be further extended to work on distributed systems. Data visualization techniques can also be incorporated in the application to get a clearer picture of the results obtained.

Acknowledgements

We are very thankful to the faculty reviewers of the department of Computer Science & Engineering for giving us valuable comments for improvising this paper.

References

[1] Olatunbosun, O.A., Edouard, L. (2021) "Evidence-Based Antenatal Care" *Okonofua, F., Balogun, J.A., Odunsi, K., Chilaka, V.N. (eds) Contemporary Obstetrics and Gynecology for Developing Countries . Springer, Cham*

[2] A. Fanelli, G. Magenes, M. Campanile and M. G. Signorini. (Sept. 2013) "Quantitative Assessment of Fetal Well-Being Through CTG Recordings: A New Parameter Based on Phase-Rectified Signal Average," *IEEE Journal of Biomedical and Health Informatics, vol. 17, no. 5, pp. 959-966*

[3] Zhang, S., Xu, H., Zhang, L., Qiao, Y. (2020) "Cervical cancer: Epidemiology, risk factors and screening" *Chinese journal of cancer research = Chung-kuo yen cheng yen chiu, 32(6), 720–728.*

[4] Yubo Fan, Yifan Meng, Shuo Yang, Ling Wang, Wenhua Zhi, Cordelle Lazare, Canhui Cao, Peng Wu. (2018) "Screening of Cervical Cancer with Self-Collected Cervical Samples and Next-Generation Sequencing", *Disease Markers, vol. 2018, Article ID 4826547, 4 pages*

[5] Moritz Kulessa, Alejandro Molina, Carsten Binnig, Benjamin Hilprecht, Kristian Kersting. (Nov 2018), "Model based Approximate Query Processing" *EDBT: Proceedings of the 22nd International Conference on Extending Database Technology*

[6] Kaiyu Li Guoliang Li. (Sep 2018) "Approximate Query Processing: What is New and Where to go?" *Springer: Data Science and Engineering*

[7] Yongjoo Park, Ahmad Shahab Tajik , Michael Cafarella, Barzan Mozafari. (May 2017) "Database Learning: Toward a Database that Becomes Smarter Every Time" *SIGMOD '17: Proceedings of the 2017 ACM International Conference on Management of Data*

[8] Sameer Agarwal,Henry Milner, Ariel Kleiner, Ameet Talwalkar, Michael Jordan, Samuel Madden, Barzan Mozafari, Ion Stoica. (June 2014) "Knowing when you're wrong: Building fast and reliable Approximate query processing systems" *SIGMOD '14: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*

[9] Yassin S. Mehanna, M. Mahmuddin and Hend S. Abdelaziz. (2015) "Approximate Query Processing Concepts and Techniques", in Joanne Evans and Lester Hunt (eds) *The Proceedings of the International Conference on Digital Information Processing, Data Mining, and Wireless Communications*

[10] Brian Babcock, Surajit Chaudhuri and Gautam Das. (2003) "Dynamic Sample Selection for Approximate Query Processing", in Joanne Evans and Lester Hunt (eds) *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*

[11] Surajit Chaudhuri, Bolin Ding and Srikanth Kandula. (2017) "Approximate Query Processing: No Silver Bullets", *Proceedings of the 2017 ACM International Conference on Management of Data*

[12] Jackson Isaac and Sandhya Harikumar. (2016) "Logistic Regression within DBMS", *2nd International Conference on Contemporary Computing and Informatics (IC3I)*

[13] Sandhya Harikumar and M.R. Kaimal. (2021) "SubspaceDB : In-database subspace clustering for analytical query processing", *Construction and Analysis of Scientific and Technological Personnel Relational Graph for Group Recognition,International Journal of Software Engineering and Knowledge Engineering*

[14] Sandhya Harikumar and Shilpa Joseph. (April 2021) "Subspace Clustering Using Matrix Factorization", *Springer, Singapore, 21*

[15] G. Veena, Peter, A. S., Rajkumari, K. A., and Ramanan, N. (2016) "A concept-based model for query management in service desks", *International Conference on Innovations in Computer Science & Engineering (ICICSE-2016) Proceedings in Springer Advances in Intelligent Systems and Computing, Guru Nanak Institutions, Hyderabad, India, 2016, vol. 413, pp. 255-265*

[16] S. Saravanan, E., K. K., Balaji, A., and S., A. (2017) "Data Classification Using Machine Learning Approach", *3rd International Symposium on Intelligent Systems Technologies and Applications (ISTA'17), Manipal University, Karnataka , 2017.*

[17] Deepa Gopakumar O. S., Ani R., Sasi, G., and Sankar, R. (2016) "Decision Support system for diagnosis and prediction of Chronic Renal Failure using Random Subspace Classification", *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Jaipur, India, 2016*