# Student Outcome Assessment on Structured Query Language using Rubrics and Automated Feedback Generation

Sidhidatri Nayak[1], Reshu Agarwal[2], Sunil Kumar Khatri[3], Masoud Mohammadian[4]
Amity Institute of Information Technology, Amity University, Noida, India[1, 2]
Research, Innovation and Extension Activities, Amity University, Noida, India[3]
University of Canbera, Canbera, Australia[4]

***Abstract*—Automated assessment of student assignment based on SQL(Structured Query Language) queries is an efficient method for evaluating and providing feedback on their DBMS-related skills. This paper provides a three step approach of how student submissions are assessed automatically using various machine learning approaches and introduced an automated grading system for SQL(Structured Query Language) queries. ASQGS (Automated SQL Query Grading System) is the process of evaluating SQL queries submitted by students of a classroom. Due to the difficulties involved in the automatic grading procedure, this endeavor continues to attract the researcher's interest in developing a new and superior grading system. The purpose of this study is to demonstrate how text relevance is calculated between a reference query that the teacher sets and a query that the student submits. To compute the grade, the similarity value between the student and reference queries will be compared. In this paper various feature similarity techniques were discussed which is required before applying the machine learning model to automatically assess the grade of the student's SQL assignment. In the second step the grade received by the ASQG is used for student outcome assessment using rubrics with respect to Bloom's taxonomy and finally scores can be calculated using predefined rubrics criteria. Additionally, in the 3rd step the system can generate feedback for students, highlighting specific areas of improvement, errors, or suggestions to enhance their queries among different groups of students segregated by their SQL knowledge.***

***Keywords*—*Automated SQL Query grading system; Cosine similarity; LSA; Multinomialnb; KNN; Logistic regression; student outcome assessment; rubric; feedback***

## I. Introduction

ASQG (Automated SQL Query Grading) is the task of assessing student's SQL (Structured Query Language) queries by leveraging computational methods. The task of ASQG can be handled with the machine learning approaches. Automatic grading has been a popular area among researchers due to the benefits of decreasing human errors and time consumed [1]. Automatic grading of SQL queries enhances advancement and improves subject learning [2]. The goal of this study is to show how text relevance computation is used while comparing a reference query with the student's query. An instructor-authored reference question is one that is written by them. To determine its similarity score, the student answer query will be compared to the reference query. Summative evaluation is used to evaluate students' effectiveness and progress in gaining comprehensive SQL knowledge. The assessment of Automated SQL Query Grading (ASAG) is more difficult since it involves a comprehension of the RDBMS idea, the schema, and a more extensive study of the search criteria. This work proposes an optimal model for autonomously grading short-answer questions using a dataset acquired from a university student taking SQL as one of their modules. ASQG deals with SQL queries that have brief replies that are frequently evaluated against a reference answer. The primary goal is to grade a learner's response regarding the model solution. Many ways to assess SQL queries do not include sentence form or coherency. It is crucial to note that automated grading systems can be configured to handle varying levels of SQL query complexity, ranging from simple SELECT statements to more advanced topics like JOINs, subqueries, and optimization techniques [3]. Instructors can save time by using automated SQL query grading. The aim of this research is to automated grading can be combined with rubrics-based assessment to analyze student outcome and generate precise feedback to enhance student knowledge in SQL.

### A. Research Methodology

Students typically submit their SQL queries through an online platform or system designed for automated grading. The platform should allow students to enter their queries and execute them against a predefined database schema. The submitted SQL queries are executed against the database schema to retrieve the results. The system compares the results obtained from executing the student's query against the expected results for matching the similarity. The expected results are typically predefined by the instructor or generated based on a reference implementation. Grading criteria are established to evaluate the correctness and quality of the queries. The comparison of instructor and student queries in ASQG may be facilitated by semantic textual similarity and paraphrasing communities [4]. This may include criteria such as accuracy, whether the model query is matching the student query and then grade is obtained as 0 or 1 based on the similarity [5]. Based on the grading criteria, the automated system assigns scores to each query submission. Scores can be calculated using predefined rubrics or algorithms which are discussed in the paper. Additionally, in step 3 the system can generate feedback for students, highlighting specific areas of improvement, errors, or suggestions to enhance their queries. The automated grading system can provide detailed error reports to help students identify and rectify the mistakes in their queries with scores to individual students and different action can be taken according to the level of the student. These reports may include syntax errors, semantic errors [6] [7] [8], or logical errors, database skill, concept and logic building skill, optimization skill the query readability through the documentation skill encountered during query execution.

## II. Dataset Preparation

According to this hypothesis, initially, the dataset was considered as an assignment comprised of student-submitted SQL queries [9]. In this section, we will collect student assignments and use them as input datasets. The dataset will be used for academic study. The dataset was created by the instructors for students from several areas at a university who were studying SQL as part of their coursework. The grading is primarily a classification issue, with class level grades being assigned as correct (1) or erroneous (0). If the student's question matches the reference query, the class level is accurate; otherwise, the class level is erroneous.

The SQL assignment is based on the conceptual diagram given in the diagram. Here the EER consisting of 4 entities. In the Employee relation each employee is uniquely identified by the primary key employee_id. Department_id is the primary key in the Department table. Location_id is the primary key in the Locations table and Job_id the primary key in the Jobs table. Each employee works in exactly one department. So, department_id is the foreign key in the Employees table referring to the department_id of the Department table. Each department is located at a particular city and location_id of department table is the foreign key referring to the location_id of the Locations table. Some employees manage other employees hence manager_id became the foreign key referring to the employee_id of the same table. Each employee's job detail is maintained in the Jobs table and job_id of Employee table is designed as the foreign key to job_id of Jobs as shown in Fig. 1.

In this model, initially, the dataset was viewed as an assignment, consisting of submitted SQL queries from students. Here, we will acquire student assignments and utilize them as input datasets. The dataset will be utilized for research purposes. A university's engineering students from a variety of fields studying SQL as part of their curriculum compiled the dataset as shown in Table I. The grading is essentially a classification problem, with class level being graded as correct (1) or incorrect (0). The class level is correct if the student's query matches the reference query, and incorrect if the student's query does not match the reference query or differs marginally from the reference query.

A Student's SQL query answer can be defined as a piece of text fulfilling the query condition [4]. A student response to a given question must be in natural language followed by the SQL syntax. A response length must be limited to between one sentence. A student response must demonstrate the external knowledge which they gained from their understanding of the Shema given and is identified within the question.

The actual dataset is prepared by collecting the student solutions for the SQL based questions through online exam conducted through google drive. The final dataset looks as follows where MA represent Model Answer and SA represents the student answer and the grade manually assigned by the instructor represented in the mark column. The word cloud for the model answer and student answer containing frequent words is represented in Fig. 2.

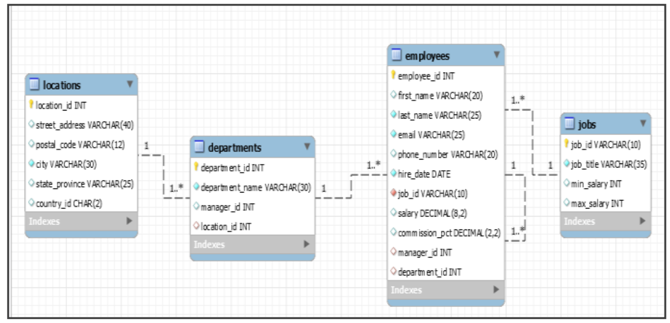


Fig. 1. The EER diagram of the company database.

TABLE I. Brief Overview of the Dataset

| **Sample Question** | ***Sample Question with model and student answer*** | | | |
|---|---|---|---|---|
| | ***Q1*** | ***Display the name of the top earner in the organization*** | ***Grade*** | ***Teacher's feedback*** |
| Model Answer for Q1 | 1 | Select first_name from employees order by salary desc limit 1; | 1 | excellent |
| Model Answer for Q1 | 2 | Select first_name from employees where salary=(select max(salary) from employees); | 1 | excellent |
| Student answer Q1 | 1 | Select first_name from employeesorder by salary desc limit 1; | 1 | excellent |
| Student answer Q1 | 1 | Select first_name from employees where salary=(select max(salary) from employees); | 1 | excellent |
| Student answer Q3 | 2 | Select first_name,max(salary) from employees; | 0 | Semantic error |
| Student answer Q3 | 3 | Select maximum(salary) from employees; | 0 | Syntax error |
| Student answer Q4 | 4 | Select max(salary) from employees; | 0 | Semantic error |

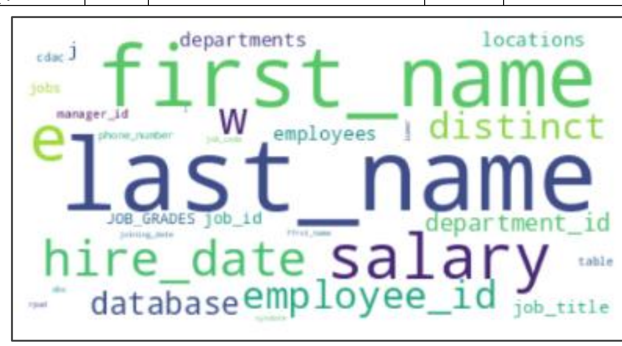


Fig. 2. The word cloud for the model answer and student answer containing frequent words.

## III. Dataset Preprocessing

The student's query may contain different forms of trash values, noisy text, and encoding. This must be cleansed for NLP to conduct additional jobs. Non-ASCII values, special characters, HTML elements, stop words, raw format conversion, and so on should all be removed during this preparation step. All sentences are switched to lower case for symmetry. We minimize some punctuation because it has no effect on the computation. The next step is to convert the two sentences into lower case as there is no difference in meaning between “create” and “CREATE” and “Create”. The third step is tokenizing the sentences. Following tokenization, each token will be compared against the terms in the user-created stop word list. Several stop words, including "in," "from," "by," "into," and "as," are used in the SQL query. Therefore, all other matching words will be eliminated aside from these stop words, leaving only the keywords for the connected phrase. We might shorten the duration to the following step by using the stop word removal. Given our consideration of syntax and semantics, the prefix-containing word need not be transformed into its root word. Therefore, stemming is not necessary for preprocessing.

## IV. Feature Similarity

We describe the proposed method for computing vector similarity. The surface closeness (lexical similarity) and significance (semantic similarity) of two "adjacent" sections of the text should be defined by text similarity [9]. Automated evaluation uses various text similarity methods to determine the similarity between two queries.

### A. String-based Similarity

Regardless of the meaning of the two strings, it examines two character sequences and determines a similarity score based on the string that corresponds to each of the two strings.The Jaccard index is commonly used to compare the similarity, dissimilarity, and distance of a data set's syntax [10]. As shown in the Eq. (1) below, the Jaccard similarity coefficient between two data sets is calculated by dividing the number of shared characteristics by the total number of properties.

I. List the unique words in the documents.
II. Find the intersection of words list of doc1 & doc2.
III. Find the union of words list of doc1 & doc2.
IV. Calculate Jaccard similarity score using length of intersection set divided by length of union set

Fig. 3. Algorithm for finding Jaccard similarity between documents.

It equals the number of unique characteristics minus the number of characteristics shared by all, divided by the total number of characteristics. Algorithm for finding Jaccard similarity between documents is in Fig. 3.

$$S(A,B) = |A \cap B| / |A \cup B| \quad (1)$$

Example1:

doc_1="select * from employee"

doc_2="select * from employee"

Jaccard_Similarity(doc_1, doc_2)

Output: 1

Example2:

doc_1="select * from employee"

doc_2="select all from employee"

Output:0.6

### B. Similarity based on Semantics

A set of terms or texts defines semantic-based similarity [9] [11]. The comparison is based on the semantic content or their coherent meanings. Semantic-based similarity makes use of the following algorithms:

*1) Similarity based on corpus:* It constructs a knowledge space using information collected just from the analysis of large corpora, which is then used to compute the connections between words and sentences [11].

*2) Latent Semantic Analysis (LSA):* LSA is a type of statistical model that uses vector means in the context of semantics for assessing the resemblance of texts or phrases [12]. LSA is a technique in NLP, to map between two documents and terms. So here the sample document consists of the model answer query and the student answer query 1, student answer query 2 and student answer query 3. So total 4 sentences. So, we need to find out which sentences are more similar. From the raw data, LSA generates a term-document matrix, which lists terms in rows and documents in columns, with each cell indicating how frequently a term appears in this document. Here each row in the matrix represents the terms in the answer query and each column represents a document or the query [13]. If the term is present, then it is represented as 1 otherwise 0. In Step 2: We are going to create a TF-IDF matrix using TfidfVectorizer. This stage transforms the text into a matrix representation, with each row representing a document and each column representing a unique word in the corpus.

Then an SVD (Singular Value Decomposition) is used to convert a large document (Query in our paper) to matrix of small size by finding the similarity between the columns and hence by reducing the number of rows. SVD helps in factorizing a complex matrix. SVD is a decomposition technique for decomposing a matrix into the constituent element. Here the matrix A is factorized into three matrices as in Eq. (2).

$$A = SU * Sy * V^T \quad (2)$$

U and V are left and right singular vectors of A respectively and S represent the singular value of A where U and V are orthogonal matrices, means if the product of a matrix and the transpose gives identity value.

$$U * \sum X \quad (3)$$

where, ∑ is a diagonal matrix containing singular values of A. A matrix is diagonal if it has nonzero elements only in the diagonal.

All have the diagonal value of ∑ denoted as $\sigma i$ and ordered as $\sigma 1 \geq \sigma 2 \geq \sigma k$ and r is the index such that $\sigma r > 0$ and either k=r and $\sigma r+1 = 0$.

Here we are going to apply Singular Value Decomposition (SVD) using TruncatedSVD to reduce the dimensionality of the TF-IDF matrix [14]. Here, we specify the number of components (dimensions) we want to reduce (in this case, 2).

Finally, the cosine angle between the vectors of the two columns is computed to compare the model query with the student query. The angle cosine between two vectors determines whether the two vectors are referring to nearly the same trend, hence cosine similarity is widely used to evaluate distance. A score close to 1 implies similarity, while a value close to 0 shows full variance. Using cosine similarity, we determined the cosine similarity between the generated LSA matrix. The resultant similarity matrix will have values ranging from 0 to 1, with higher values representing greater similarity between documents. The similarity matrix is then printed to the console. So, we have implemented LSA for document similarity in Python using the scikit-learn library.

### C. *Vector-based Similarity*

Vector-based similarity techniques in Natural Language Processing (NLP) involve representing textual data as numerical vectors and using various similarity measures to determine the similarity or relatedness between different texts. These techniques are widely used for tasks such as document similarity, semantic search, clustering, and information retrieval. Here are some common vector-based similarity techniques in NLP:

*1) Bag-of-Words (BoW) Model:* Each text is encoded as a vector in this manner, with each dimension corresponding to a distinct word in the lexicon. Each dimension's value shows the frequency or occurrence of that term in the manuscript.

*2) Term Frequency-Inverse Document Frequency (TF-IDF):* The TF-IDF approach is well-known for assigning weights to words in a document based on their frequency in the document and inverse frequency throughout the whole corpus. Each page is represented as a vector of TF-IDF scores, and cosine similarity or other distance metrics can be used to determine similarity [15][16][17].

## V. Machine Learning Approach for Automated SQL Query Grading

Machine learning can be effectively used in automated short answer grading systems to streamline the process of evaluating and providing feedback on student responses 18][19]. This paper deals with an approach for building a machine learning system in python that uses K-Nearest Neighbors (KNN), multinomialNB and logistic regression method for the classification of textual documents for the dataset discussed above individually. The best model can be chosen to find the grade of the student. The primary difficulty in characterizing texts is that they are an assortment of letters and words. We require a numerical representation of those words to input them into our models, which will compute distances and make predictions. Bag of words and tf-idf are two methods for numerical representation [20] [21]. The experimental phase of the investigation was carried out using textual documents taken from the dataset's model answer and student answer. Prepare a labeled dataset of short answers, where each answer is associated with a grade or score. You'll need a set of answers that are already graded by humans.

*1)* Preprocess the short answers by removing punctuation, converting all letters to lowercase, and applying any other necessary preprocessing steps like stemming or lemmatization. This step helps standardize the text data.

*2)* Convert the preprocessed short answers into numerical features that model can work with. One common approach is to use the term frequency-inverse document frequency (TF-IDF) representation. This representation assigns weights to each word based on its frequency in a specific short answer and across the entire dataset. Because we must vectorize both the model answer and the student answer separately. After vectorizing the model answer and student query we concatenate the two vectors to create the train and test vectors as follows.

*3) Train-Test Split:* Split your dataset into a training set and a test set. The training set will be used to train the classifier, while the test set will be used to evaluate its performance.

We have used various machine learning models to automate the grading process with following result.

### D. *KNN Classification for Automated SQL Query Grading*

The projected class label in KNN classification is decided by voting for the nearest neighbors, that is, the majority class label in the set of the selected k examples is returned. [23][24][25]. The quality of the predictions depends on the distance measure. We use cosine as distance measurement technique. Therefore, the KNN algorithm is suitable for applications for which sufficient domain knowledge is available. We have used the K Nearest-Neighbors Classifier method of sklearn. Neighbors class. Fit the k-nearest neighbors' classifier from the training dataset. After that we can use the predict () to predict the class for the test data. And the accuracy of the model is 70%, followed by the classification report in Fig. 4 and confusion matrix in Fig. 5.

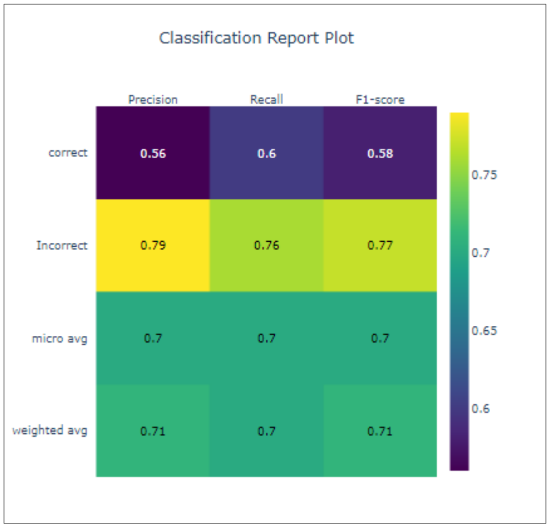


Fig. 4. Classification report of class prediction using KNN algorithm.

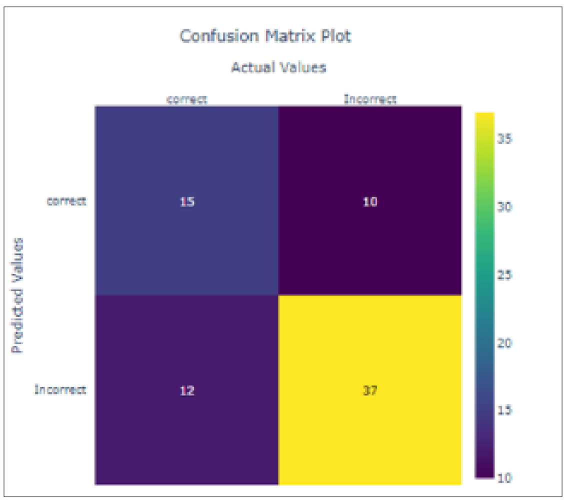


Fig. 5. Confusion matrix of class prediction using KNN algorithm.

## E. B. *Multinomial Naïve Baye's Classification for Automated SQL Query Grading*

Multinomial Naive Bayes classifier is a specific instance of a Naive Bayes classifier that employs a multinomial distribution for each of the features. Multinomial Naive Bayes assumes multinomial distribution for all pairings, which is a reasonable assumption in certain circumstances, such as for word counts in documents [22]. Multinomial Naive Bayes (MNB) can be used for automated short answer grading tasks. MNB is a popular algorithm for text classification tasks, including student's SQL query grading, because it can handle multiple classes and works well with discrete features like word counts. Here's how you can use MNB for automated short answer grading:

*1) Model training:* Train the MNB classifier using the training set and the corresponding grades. MNB calculates the probability of a short answer belonging to a particular grade based on the occurrence of words in the answer.

*2) Model evaluation:* Evaluate the performance of the trained MNB classifier using the test set. You can use metrics like accuracy, precision, recall, or F1 score to assess how well the classifier is grading the short answers.

*3) Grading new answers:* Once the MNB classifier is trained and evaluated, you can use it to automatically grade new sort answers from the test data. We observe Multinomialnb with 84% accuracy and followed by the classification report in Fig. 6 and confusion matrix in Fig. 7.

It's important to note that MNB is a simple and fast algorithm, but it has certain assumptions, such as the independence of features. While MNB can work reasonably well for short answer grading tasks, more advanced machine learning algorithms or natural language processing techniques might be required for more complex grading scenarios.

## F. *Logistic Regression in Automated SQL Query Grading*

Logistic regression is a classification algorithm commonly used in machine learning to predict binary outcomes. While it may not be directly applicable to automated SQL query grading, logistic regression can be used as part of a broader approach to assess the quality or correctness of SQL queries [18]. We train a logistic regression model using the labeled training dataset and the extracted features. Python provides various machine learning libraries such as scikit-learn or TensorFlow that offer easy-to-use implementations of logistic regression. Evaluate the trained logistic regression model using the test set. Assess the model's performance metrics such as accuracy, precision, recall, or F1-score to determine how well it predicts the correctness of SQL queries. Once the logistic regression model is trained and evaluated, you can use it to grade new, unseen SQL queries.

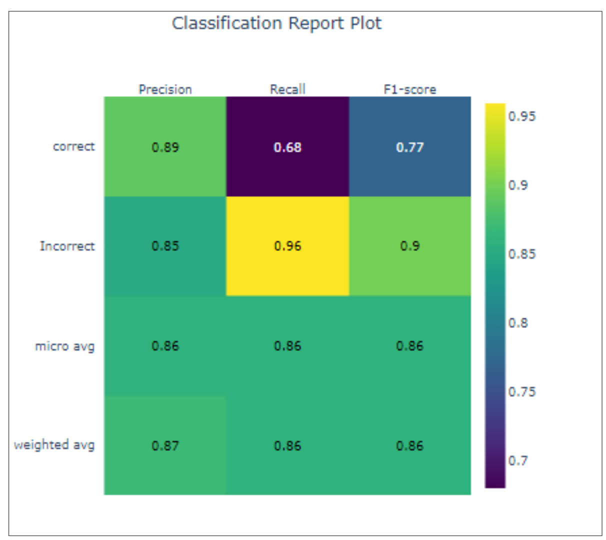


Fig. 6. Classification report of class prediction using multinomial Naïve Baye's algorithm.

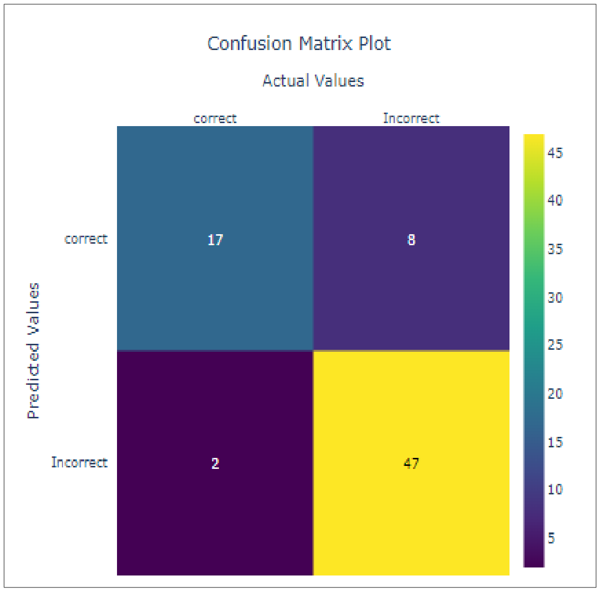


Fig. 7. Confusion Matrix of class prediction using multinomial Naïve Baye's algorithm.

Extract the features from the new queries and pass them through the trained model to obtain the predicted probabilities or classes (correct or incorrect) for each query. It's worth noting that logistic regression alone may not be sufficient for comprehensive SQL query grading. You may need to incorporate other techniques, such as natural language processing (NLP) or more complex machine learning algorithms, depending on the specific grading criteria and requirements of your system and the plot classification is as presented on the Fig. 8 and Fig. 9.

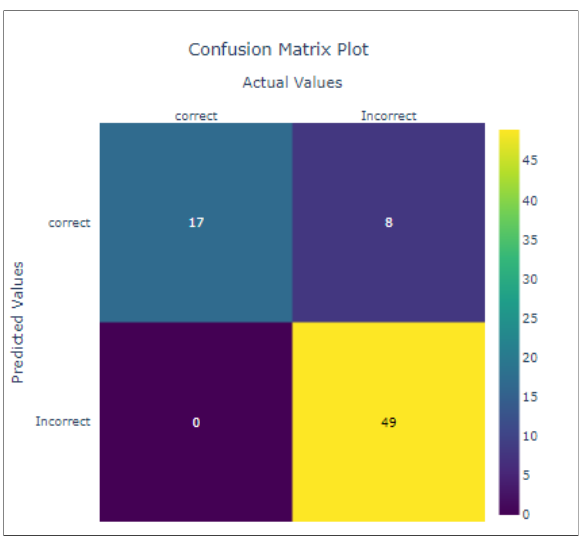


Fig. 8. Confusion matrix for the class prediction using logistic regression.

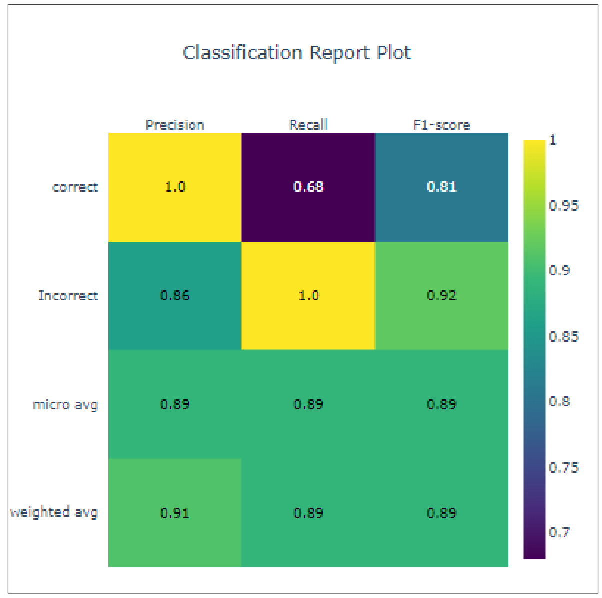


Fig. 9. Classification report for the class prediction using logistic regression.

Out of the three machine learning algorithms, the Logistic Algorithm shows better results compared to the other two machine learning algorithms in finding the grade of the student. So, in the next step this grade can be used for student outcome assessment using predefined rubrics by the author.

## VI. Student Outcome Assessment Using Rubric

Rubrics are scoring guides that outline specific components and expectations for an assignment. Instructors were required to utilize outcome-based assessment methods and methodologies to evaluate students' learning against predetermined outcomes [26]. In this paper, rubrics approach is used for assessment of SQL query assignments submitted by the students in the university. A rubric is a tool that empowers students to guide their own learning process. It is an effective technique for being learner-centered [27]. Rubrics for assessment can improve consistency, save time in grading, provide timely feedback, promote student learning, clarify expectations, and refine teaching methods. Rubrics assist students in understanding assignment objectives, gaining awareness of their learning progress, and receiving timely and detailed feedback to enhance work. Rubrics measure students' achievement of learning outcomes, not their performance in comparison to peers [28].

This research's second goal is to create a scoring system for SQL query assignments. and provide feedback to the student using clustering techniques of unsupervised machine learning algorithms. Along with ASQG (Automated SQL Answer Grading), discussed in part 1 of this paper, instructor can design the rubrics for assessment of student assignments and student outcome analysis. ASQG mostly focuses on the assessment of SQL query by classification of student submission as correct or not correct after comparing with the model answer. But the SQL (Structured query language) query can be assessed by several other parameters using rubrics. The instructor assistant rubrics can be created for assessment of various other criteria mentioned below for student learning outcome assessment and scoring strategy and performance descriptor. The criteria that have been chosen are grounded in Bloom's Taxonomy of Learning Domains. The student learning outcome can be assessed in to six levels such as, remember, understand, apply, analyze, and evaluate [29] [30].

*1)* Theory and conceptual knowledge on database and SQL like database design, unique and referential integrity constraints, concept of normalization, functional dependency, ER diagram etc.

*2)* The strong knowledge in SQL (Structured query language) can be measured by student's query syntax, use of DDL (Data Definition Language), DML (Data Manipulation Language), various keywords and clauses in correct order to get a correct output.

*3)* Conceptual thinking skill can be a parameter to assess student's learning outcomes.

*4)* Similarly, the logic building ability can be used to assess student's in-depth knowledge in SQL.

*5)* The efficiency of the SQL query can be checked with optimal performance like low query execution time, minimal resource consumption.

*6)* The assignment should be well documented by following instructions, better query readability and proper comment to explain the work.

A rubric for SQL assignment is developed in this paper. The above criteria (1-6) are used to assess the student's skill in SQL which is listed in Table II. Each criterion in this rubric has a four-point grading system in the following manner.

<40% score indicates very poor knowledge in the criterion and needs development.

40-59% score indicates limited knowledge in the criterion and still needs development.

60-79% score indicates adequate knowledge in the criterion and need practice.

80-100% score indicates outstanding knowledge.

TABLE II. RUBRICS FOR STUDENT OUTCOME ASSESSMENT

| *Category* | *Assessment Criteria* | *Poor <40%* | *Good 40-59%* | *Very good 60-79%* | *Excellent 80-100%* | *Learning level based on Bloom taxonomy* |
|---|---|---|---|---|---|---|
| ***Theory and concept knowledge on database and SQL*** | ***Design of relation with integrity constraints*** | Basic knowledge of database structure and design without key concept | Concept is not very clear, but tables are partially created | Good understanding of the concept with minor error | Clear and logical concept of relational database, well designed tables with appropriate data types and relationships | Remember |
| | ***Normalization*** | Tables not normalized | Some normalization but with significant issues Table | mostly normalized with minor issues | Properly normalized tables without errors | understand |
| ***SQL query knowledge*** | ***syntax*** | Queries do not execute with major syntax errors | Queries do not execute with noticeable syntax errors | Queries do not execute with minor syntax errors | Queries are well-structured with correct syntax | Remember |
| ***Conceptual thinking and skills*** | ***Data retrieval from the table*** | incomplete data retrievals | partially correct with minor logical issue | Logic is clear but output not as expected | Accurate logic and complete data retrieval as expected | remember |
| | ***Filtering and sorting*** | Unable to apply, incorrect output | partially applied with errors | Mostly applied correctly | Accurately applied | remember |
| ***Critical thinking and logic building*** | ***Joins*** | Incorrect concept of join | Missing join conditions with ambiguous result | Joins applied but incorrect output | correctly applied with accurate output | Understanding and apply |
| | ***Aggregate functions*** | Incorrect use of aggregate functions | Missing join conditions with ambiguous result | Joins applied but incorrect output | correctly applied with accurate output | analyze |
| | ***subquery*** | Incorrect use of | Partially correct | Mostly syntax correct but logical error | Correct output | analyze |
| ***Efficiency*** | ***Query Performance*** | Queries execute very slowly or not at all | Queries execute with noticeable delay | Queries execute with minor delay | Queries execute efficiently without delay | Evaluate |
| | ***Indexing*** | No indexing implemented | Indexing partially implemented with minimal impact | Appropriate indexing implemented | Optimal indexing implemented for performance | Evaluate |
| ***Documentation*** | ***Readability*** | Query and database structure are unreadable | Query and database structure are somewhat readable | Query and database structure are mostly readable | Query and database structure are highly readable | Create |
| | ***Instructions*** | Instructions not followed | Partially followed instructions with notable deviations | Mostly followed instructions with minor deviations | Instructions fully followed | create |

TABLE IV. GRADES OF SAMPLES OF 10 STUDENTS IN RANDOM AS PER THE RUBRICS AND ASQG

| Sl. No. | Theory &concept | SQL syntax | Query skill | Logical skill | Efficiency | Doc | Mean |
|---|---|---|---|---|---|---|---|
| 1 | 70 | 90 | 100 | 80 | 50 | 100 | 82 |
| 2 | 80 | 90 | 100 | 80 | 60 | 100 | 85 |
| 3 | 80 | 60 | 60 | 50 | 40 | 60 | 58 |
| 4 | 60 | 70 | 70 | 60 | 50 | 80 | 65 |
| 5 | 40 | 50 | 60 | 50 | 40 | 60 | 50 |
| 6 | 90 | 100 | 10 | 90 | 50 | 100 | 90 |
| 7 | 70 | 90 | 100 | 80 | 50 | 100 | 82 |
| 8 | 30 | 90 | 100 | 80 | 50 | 100 | 37 |
| 9 | 60 | 70 | 70 | 60 | 50 | 80 | 65 |
| 10 | 70 | 80 | 90 | 80 | 60 | 100 | 80 |

Each criterion's questions were selected to evaluate the related skills. The assessment had sixty questions pertaining to the standards, 10 from each criterion. The ASQG (Automated SQL Query Grader) each answer query with 1 point for the correct answer and 0 point for the incorrect answer. For instance, if one student answered 8 correct questions from the 10 questions of a criterion, means that with 80% correct. Hence will be graded as excellent. Similarly, if one student answered three correct questions from the 10 questions of a criterion, with 30% correct attempt and gets an unsatisfactory grade. According to the preceding criteria, this partial grading method extracts the learners' competency level in each category. The correctness score of 10 students in random is presented in Table III. For example, the 1st student has answered seven questions correctly so get 70% score in 1st criteria and excellent score in SQL syntax and querying skill, the logic building, and analytical skill is also excellent, but the queries are average optimized and but 100 marks for query readability. So, on average he got 82% grade in the SQL assignment. Similarly, the table represents the final score of 10 students' data of total 150 students and 60 questions, 10 questions from each criterion used in Rubric. In the next part of the paper, we will provide feedback to the student along with score obtained in the assignment.

## VII. FEEDBACK GENERATION FROM THE RUBRICS

Rubrics assist instructors provide constructive input to students by highlighting strengths and faults and identifying spaces for improvement. Breaking down the assignment into distinct criteria and offering prompt feedback on students' strengths and weaknesses in each category provides precise information. Feedback to students on how successfully or poorly they completed an assignment. Rubrics can save time on grading assignments and provide timely feedback to students about their performance. Rubrics can be used by instructors to emphasize the various levels of expectations they have for students by establishing specific evaluation criteria for task completion. Evaluation criteria are the characteristics instructors assess when judging the quality of a student-completed job. Every component of the evaluation criteria is discussed in detail so that students understand what precise abilities, knowledge, or strategies they must possess to get a given score or grade. Rubrics can also help instructors clarify the implicit expectations for a specific task. Rubrics can reduce grading time by allowing professors to award specific scores instead of lengthy comments for each work. Thus, rubrics can be utilized to evaluate students' work in a more efficient and transparent manner. Rubrics can help teachers justify a student's score or grade to other participants, including parents and university authorities.

Although the feedback provided by the rubric is sufficient for individual student input, we have used a novel method in this research to separate the students based on their learning competencies. This allows the instructor to divide the students into groups based on the rubric score. Here in this paper, we use the k-mean Clustering technique, an unsupervised learning process, can be used to separate students into clusters in order to study class patterns and assess querying ability. Clustering can assist in identifying clusters in which students in one cluster have nearly the same levels of knowledge and thus can receive similar feedback and improvement advice. Students in the cluster with limited knowledge can receive further assistance to enhance their skills. Students in the cluster where all students have excellent SQL knowledge should be encouraged to focus on advanced topics and application design in real-world use cases.

## VIII. CONCLUSION

In this research, Students typically submit their SQL queries assignment on an online platform. Students write queries and execute them against a predefined database schema. The ASQG compares the results obtained from executing the student's query against the expected results for matching the similarity. The expected results are typically predefined by the instructor or generated based on a reference implementation. Grading criteria are established to evaluate the correctness and quality of the queries. The comparison of instructor and student queries in ASQG may be facilitated by semantic textual similarity and paraphrasing communities. This may include criteria such as accuracy, whether the model query is like the student query. Based on the grading criteria, the automated system assigns scores to each query submission. Scores can be calculated using predefined rubrics or algorithms which are discussed in Section VI of the paper. Additionally, in section VII the system can generate feedback for students, highlighting specific areas of improvement, errors, or suggestions to enhance their queries. The automated grading system can provide detailed error reports to help students identify and rectify the mistakes in their queries with scores to individual students and different action can be taken according to the level of the student. These reports may include syntax errors, semantic errors or logical errors, database skill, concept and logic building skill, optimization skill the query readability through the documentation skill encountered during query execution.

## REFERENCES


[1] Haller, S, Adina A, Christin, S., and Nicola S. "Survey on automated short answer grading with deep learning: from word embeddings to transformers.",2022, 1(1), pp. 1-29.

[2] Nayak, S., Agarwal, R., & Khatri, S. K. “Review of Automated Assessment Tools for grading student SQL queries”. International Conference on Computer Communication and Informatics (ICCCI),2022 (pp. 1-4). IEEE.

[3] Ahadi, Alireza, Vahid Behbood, Arto Vihavainen, Julia Prior, and Raymond Lister. "Students' syntactic mistakes in writing seven different types of SQL queries and its application to predicting students' success." In Proceedings of the 47th ACM Technical Symposium on Computing Science Education, .(2016).pp. 401-406.

[4] Burrows, S., Gurevych, I. and Stein, B., “The eras and trends of automatic short answer grading”. International journal of artificial intelligence in education, 25, (2015), pp.60-117.

[5] Suzen, N., Gorban, A.N., Levesley, J. and Mirkes, E.M.,” Automatic short answer grading and feedback using text mining methods”. Procedia computer science, 169,(2020) pp.726-743.

[6] Welty, Charles. "Correcting user errors in SQL." International Journal of Man-machine studies 22, no. 4 (1985): 463-477.

[7] Buitendijk, R. B. "Logical errors in database SQL retrieval queries." Computer Science in Economics and Management 1 (1988),pp. 79-96.

[8] Al-Salmi, Aisha. Semi-automatic assessment of basic SQL statements. Diss. Loughborough University, 2019.

[9] M. Mohler, R. Bunescu, and R. Mihalcea, “Learning to grade short answer questions using semantic similarity measures and dependency graph alignments,” in Proc. 49th Annu. Meeting Assoc. Comput. Linguistics: Hum. Lang. Technol., vol. 1, 2011, pp. 752–762.

[10] Fletcher, S., and M. Z. Islam. “Comparing Sets of Patterns with the Jaccard Index”. Australasian Journal of Information Systems, Vol. 22, Mar. 2018, doi:10.3127/ajis. v22i0.1538.

[11] Gomaa, Wael H., and Aly A. Fahmy. "Short answer grading using string similarity and corpus-based similarity." International Journal of Advanced Computer Science and Applications (IJACSA) (2012).,3, no. 11.

[12] R. Klein, A. Kyrilov, and M. Tokman, “Automated assessment of short free-text responses in computer science using latent semantic analysis,” in Proc. 16th Annu. Conf. Innov. Technol. Comput. Sci., (2011), pp. 158–162.

[13] Thomas, N. T., Ashwini Kumar, and Kamal Bijlani. "Automatic answer assessment in LMS using latent semantic analysis." Procedia Computer Science 58 (2015): pp. 257-264.

[14] Cline, Alan Kaylor, and Inderjit S. Dhillon. "Computation of the singular value decomposition." Handbook of linear algebra. Chapman and Hall/CRC, (2006). pp.45-1.

[15] Zhang, Wen, Taketoshi Yoshida, and Xijin Tang. "A comparative study of TF* IDF, LSI and multi-words for text classification." Expert Systems with Applications 3.38 (2011): pp. 2758-2765.

[16] Faruqui, M., Yulia, T., Pushpendre, R. and Chris, D. “Problems with evaluation of word embeddings using word similarity tasks”, 2016, Proceeding of the 1st workshop on evaluating vector-space representations for NLP, pp. 30-35.

[17] Lubis, Fetty Fitriyanti, et al. "Automated Short-Answer Grading using Semantic Similarity based on Word Embedding." International Journal of Technology,12(3) 2021, pp. 571-581.

[18] Vaswani, A. et al. "Attention is all you need." Advances in neural information processing systems”, Proceedings of 31stConferenceonNeural Information Processing Systems, 2017, LongBeach,CA,USA.

[19] Basu, Sumit, Chuck Jacobs, and Lucy Vanderwende. "Powergrading: a clustering approach to amplify human effort for short answer grading." Transactions of the Association for Computational Linguistics 1 (2013),pp.391-402.

[20] Ghavidel, Hadi Abdi, Amal Zouaq, and Michel C. Desmarais. "Using BERT and XLNET for the Automatic Short Answer Grading Task." In CSEDU (1), (2020),pp. 58-67.

[21] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter,” (2019), pp. 2-6.

[22] Galhardi, Lucas Busatta, and Jacques Duílio Brancher. "Machine learning approach for automatic short answer grading: A systematic review." In Advances in Artificial Intelligence-IBERAMIA 2018: 16th Ibero-American Conference on AI, Trujillo, Peru, November 13-16, 2018, pp. 380-391. Springer International Publishing.

[23] Trstenjak, Bruno, Sasa Mikac, and Dzenana Donko. "KNN with TF-IDF based framework for text categorization." Procedia Engineering 69 (2014), pp.1356-1364.

[24] Boisvert, Charles, and Konstantinos Domdouzis. "A comparative analysis of student SQL and relational database knowledge using automated grading tools." In International Symposium on Computers in Education (SIIE), IEEE(2018), pp. 1-5.

[25] Huang, Yuwei, Xi Yang, Fuzhen Zhuang, Lishan Zhang, and Shengquan Yu. "Automatic Chinese reading comprehension grading by LSTM with knowledge adaptation." In Advances in Knowledge Discovery and Data Mining: 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, (2018) pp. 118-129. Springer International Publishing.

[26] Gupta, B. L., and Pratibha Bundela Gupta. "A CRITICAL STUDY ON THE USE OF RUBRICS IN TECHNICAL INSTITUTIONS OF INDIA." Indonesian Journal of Education Assessment 4.2 (2021): pp. 20-33.

[27] David C. Leader. (2018). Student Perceptions of the Effectiveness of Rubrics. Journal of Business and Educational Leadership, 8(1), pp. 86-99.

[28] Chowdhury, Faieza. "Application of rubrics in the classroom: A vital tool for improvement in assessment, feedback and learning." International education studies 12.1 (2019): pp. 61-68.

[29] Chandio, Muhammad Tufail, Saima Murtaza Pandhiani, and Rabia Iqbal. "Bloom's Taxonomy: Improving Assessment and Teaching-Learning Process." Journal of education and educational development 3.2 (2016): 203-221.

[30] Priyanka Gupta, and Deepti Mehrotra. "Objective Assessment in Java Programming Language Using Rubrics." Journal of Information Technology Education: Innovations in Practice 21 (2022): 155-173.