**Abstract**

Due to the pandemic spread of COVID-19, the only substitute of delivering education was the online teaching mode. A large number of schools, colleges and universities had to move on online teaching mode. In this regard, there was a need to investigate the quality of education by evaluating the performance of teacher. There is also need to investigate the quality o education in term of student assessment. Here, we proposed a framework based on historical dataset to evaluate the performance of teacher and student by using different parameters. The ultimate goal of this study is to identify the most correlated parameters of teacher and student assessment.

**Introduction**

During the pandemic of COVID-19, the schools and universities around the change their education style from physical to online mode. Many countries have launched policy and support to enhance online teaching from primary to university education. A variety of online platforms have provided appropriate functions and tools to help teachers to conduct online teaching, such as Zoom, Skype, Blackboard Learn etc. Online education has become the only substitute for many universities in this special circumstance. In this regard, there is need to analyze the quality and effectiveness of the online offered course. The delivery of online course through digital platforms, enables the authorities to store the historical data that can be used for different analysis tasks. The collection of historical data never easy in physical learning mode. Resultantly, the delivery of education via online courses opened the new opportunities. In particular the evolution of data analytics and machine learning allows these data to be collected, analyzed and possibly identify areas where improvements can be made.

In this project we used the historical data of online education for the assessment of students and teachers score evaluation. The goal of the proposed work is to find the most correlated attributes of data for the accurate evaluation of teachers and students. There may be some attributes that are not correlated with the final grades of the students like an assignment was much easy or much difficult and not correlate with the final gradings of the student. Here, the aim of the proposed study is to analyze the matrices with different features to find out the most correlated features for the assessment of student and teacher gradings.

**Design**

**Dataset**: In the proposed problem, we taken the dataset differently for the teachers and students. The dataset contained the four features for teacher data and 12 features for student data. The teacher’s dataset was based on the feature “Student Feedback for Teacher”, “Student Online Time”, and “Student attendance” for the prediction of the teacher score. While the student dataset was based on the features of “Teacher Remarks”, “Attendance in course”, “Course Access”, “Resource Visit”, “On time Submission”, “Exam # 1”, “Exam # 2”, “Exam # 4”, “Project”, “Assignment # 1”, “Assignment # 2” and “Assignment # 3” for the prediction of student’s grades. The teacher and student dataset were consisted of 12000 and records. Further, the teacher and student datasets were categorized on the basis of feedback value (“Excellent, Good, Average, and Bad) to see the distribution of both datasets. The distribution of feedback feature reveal that both datasets were nearly equally distributed (Figure 1 and 2). The detail of each feature in student and teacher dataset is mentioned in Table 1.

Table 1: Details of features in teacher and student dataset.

|  |  |  |  |
| --- | --- | --- | --- |
| Students Dataset | | | |
| Attribute | **Detail** | **Attribute** | **Detail** |
| Teacher feedback | Teacher’s feedback related to student performance | assignment1 | Marks gain in assignment 1 |
| Attendance | Number of classes that student attend | Assignment2 | Marks gain in assignment 2 |
| Course access | How many times student access the course. | Assignment3 | Marks gain in assignment 3 |
| Resource visit | How many times student access the material related to course | Exam1 | Marks gain in Exam 1 |
| submission | How many times student submit assigned work on time? | Exam2 | Marks gain in Exam 2 |
| project | Student final score in semester project | Exam3 | Marks gain in Exam 3 |
| Teachers Dataset | | | |
| Attribute | **Detail** | **Attribute** | **Detail** |
| Student Feedback | Feedback of student related to teacher. | Attendance | Number of attended students during the class. |
| Online Time | How much time students had online during lecture? | Question | How many questions were asked by students? |

**Preprocessing:** In the processing of selected dataset, we plotted the scatter plot for each column with the targeted column (teacher score for teacher dataset and student grade for student column) in Figure 3 and Figure 4. The scatter plot showed the relatedness of the features with targeted column. Further, the correlation for all features with targeted features were calculated one by one. After the understanding of the dataset, the both datasets were passed through the cleaning data pipeline. Firstly, all the null and undefined values were filled with average value of that feature. Further, all the string value features were converted into numerical features by using label-encoder function of Sklearn library in python. The label encoder function converted the Excellent, Good, Average and Bad value in Feedback feature into 1, 2, 3, 4 numeric values respectively. Then we divided the both dataset into two categories: training set and testing set by using train-test-split function of Sklearn. Train-test-split function split the data with 70% and 30% ratio in training and testing set respectively. After the categorization of both datasets into training and testing set, the training set had the 8040 records while the testing set had the 3960 records.

|  |  |
| --- | --- |
| Figure 1: Dataset distribution of students | Figure 2: Dataset distribution of teachers. |
| Figure 3: Scatter plots of student features. | Figure 4: Scatter plots of teacher features. |

**Machine Learning**: Next initialize the well-known machine learning models including Linear Regressor, Random Forest regressor, KNN regressor, and Decision Tree Regressor for the teacher and student score prediction. All the configured ML models were applied on the training set for training of the models. The evaluation of the models was performed using test set. Numerous evaluation measures were used for calculating the performance of the models including the R2 Score, Mean Squared Error, Mean Absolute error and the Root Mean Squared error. This evaluation measured plays an important role to demonstrate the performance of the trained model. All the trained models and label encoders were saved using the pickle library of python to make predictions in future.

**Implementation**

For the development of student teacher assessment system, we configured an environment in python version 3.8 with essential libraries like Numpy, Pandas and Sklearn. For the machine learning related tasks, the built-in libraires including Sklearn, matplotlib, Scikit-learn and NLTK were installed and used in configured environment. Lastly, the user interface was integrated with the assessment module by using the web-based Django framework.

Firstly, the historical records of teachers and student during online classes were collected from different sources. The data was passed through cleaning process using NLTK and Pandas library to fill Null values and missing value. Later on, different machine learning models were trained for the accurate assessment of teacher and student Grade. The Decision Tree regressor model performed well among the all-trained model. The performance of the model was evaluated by using the evaluation measures on testing data.

The historical data was split into two sets: training set and testing set. Training set was used for the training of the machine learning models. The random state was set to 0 as hyper parameter and the rest of the hyper parameters were used with default value. The model was trained with different features of the student and teachers to find out the most correlated features. All the ML model find out the 12 features of student as most correlated feature and outer performed on these features in term of Mean Squared Error (MSE). While the 4 features of teachers were identified as most correlated features and Decision Tree Regressor outer performed on these features in term of MSE. The training model showed the 7.52 and 6.83 MSE score for the testing data of teacher and student respectively. The detailed report of evaluation measures of all the trained models for teacher score and student score is presented in table 1 and table 2 respectively.

Lastly, the interface was design using the HTML, CSS and JavaScript for the interaction of user with proposed model. The user interface was design with the key points keeping in mind like user friendly interface and interactive user interface. Later the user interface was integrated with the machine learning model by using the web-based Django framework. Django is a web-based secure framework that handle the user requests over the http protocol. Our interface was consisting of three screens: first was the home page that contain the redirect link of student assessment page and teacher assessment page. Teacher and student assessment page have the user interaction forms that take the parameters value from user for the prediction of respective Grade. On the teacher and student assessment page a submit button followed by the user form that allowed the submission of user data. After the submission of form data, the data is passes to machine learning models that take data as input and predict the Grade of respective entity. After predicting the Grade, the grade was return in JSON format to HTML page, where the AJAX call received the JSON response and showed the predicted Grade in the left panel of the page.

**Testing**

The proposed framework was tested in different ways to ensure the reliability and integrity of the system. Black Box testing, unit testing, integrated testing, code level testing and interface testing was performed on the testing set of the historical data. In the black box testing, all the examples were passes through the code and made prediction on all the examples without the understating of backend technique. The predicted values were compared to the original values and find out the MSE score of the model. The black box testing was further categorized into unit testing and integrated testing. In the Unit testing, all the written functions were tested separately. The selected examples were chosen from the test set and passed to the first function. The output of the first function was compared with the desired output. The output of function A was passed to next function B as input and the out of function B was passes next. All the functions were tested independently and compared the output of each function with desired/original/know output. In the integrated testing, all the functions were tested collectively. The whole framework was tested by giving raw input and comparing the final output with original output.

In the Code level testing, python script was tested by the terminal by running the all function. The input vectors were passed by the terminal and received the prediction score on terminal for investigation. Lastly, the framework was tested by the interfacing testing. In the Interfacing testing, the testing examples were chosen from the test set. The student and teacher assessment form were filled for each value one by one and get the score. The trained model showed the 7.52 and 6.83 average MSE score for testing set for all categories of testing. The detailed report of evaluation measures of all the trained models for teacher score and student score is presented in table 2 and table 3 respectively.

Table 2: Evaluation Measures report of teachers test set.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | R2 Score | RMSE | MSE | MAE |
| Linear Regressor | 0.917555 | 7.727267 | 59.71066 | 6.362078 |
| Random Forest Regressor | 0.92693 | 7.274672 | 52.92085 | 6.009769 |
| K-Nearest Neighbor | 0.928033 | 7.219567 | 52.12214 | 5.948182 |
| Decision tree Regressor | 0.877154 | 9.432427 | 88.97068 | 7.5246 |

Table 3: Evaluation measure report of student test set.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | R2 Score | RMSE | MSE | MAE |
| Linear Regressor | 0.930227 | 6.193569 | 38.36029 | 5.268767 |
| Random Forest Regressor | 0.936384 | 5.913984 | 34.97521 | 5.094838 |
| K-Nearest Neighbor | 0.928191 | 6.283286 | 39.47969 | 5.295505 |
| Decision tree Regressor | 0.872907 | 8.359081 | 69.87424 | 6.832828 |

**Discussion**

In the proposed study, we trained the different machine learning models for different features of the students and teachers. The identification of most correlated features for the better MSE of the machine learning model was a biggest challenge. Sometimes, teachers are unnecessarily set the paper or assignment as very easy and very hard. So, the impact of that exam and assignment can’t directly translate the final grade of the student. Like the assignment 1, assignment 2 and assignment 3 in our study are less correlated with the final output score of the students. The calculation of correlation score required the mapping of each feature against the final score that is very difficult and time consuming for large amount of data. Secondly the historical data for the teacher and student score evaluation is no longer available publicly. The generation of the student and teacher historical data was the main challenge that face during the implementation of the proposed study. The quality assurance of collected data on the basis of quality parameters is further challenging.

Here, we generated our own historical data for the teacher and student final grade prediction. The proposed study contributes in the final assessment of the student score and teacher score for online offered courses. Although, we use correlation calculation technique for the selection of features, but our dataset was limited to few features. The dataset can be updated by adding more features of teachers and students that may or may not be correlated with the final score. However, future studies can update the dataset with more features to more accurately predict the final grades. Moreover, for the large set of features data, the feature selection approach can be used for the selection on best features for ML models after correlation process. For the limited features, the proposed study performs well and the MSE scores showed that the model is robust enough to deploy in real world environment.