Abu Dhabi Men's College

Business Solution

Project Title:

MOClog

Ali Mohamed Alaryani – H00323098

Khaleefa Kulaib Alhameli - H00323136

Dr. Alexandros

Table of Content

Contents

Table of Figures

Table of Tables

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.

Chapter 1: Introduction

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.

Objective

- The first problem is student centered, where students would be eager to know how their current performance translates to their overall course grade. This would help students reflect on their current approach and maybe modify it if needed.
- The second problem is faculty centered, where the faculty would find valuable metrics on each of the performance criteria, they are using to identify student performance. This way they can intervene with weak students early on and help them improve their performance.

Also, if a performance metric does not correlate with student final performance, is this metric necessary to begin with?

- The problem might be different. If an exam is important, but the teacher is making it unnecessarily easy or difficult, then it might also not correlate well with final grades, prompting the faculty to review their exam style.

We can combine these into one encompassing problem as follows: "Analyze various student performance metrics against student course grades, and identify for each course which metrics are better indicators of student performance".

Literature Review

Several research investigations have been undertaken in the academic arena, but their work is limited to sentiment analysis (classifying sentences as good, bad, or neutral). In one study, supervised algorithms were employed to do sentiment analysis on student assessment on a social media page [1]. The goal of this research was to determine feedback direction in order to evaluate a teacher's performance. Teaching senti-lexicon [2] was created by researchers in one of the studies and comprised of a teaching corpus, a category, and a sentiment weight score. They used the SVM, ID3, and Naive Bayes algorithms to test their theories. They came to the conclusion that teaching senti-lexicon rather than generic lexicon produced satisfactory outcomes (SentiWordNet). In [3], sentiment analysis was used to predict a student 's educational failure by analyzing their self-evaluated comments. As a result, the teacher will be able to improve the teaching process even more. Another research work presents a random forest algorithm [4] to assist in the creation of annual appraisal reports. They generated a report that took into account not only student comments but also numerous performance criteria. In [5], proposes a lexicon-based sentiment analysis system that generates sentiment analysis-based metrics that appear to be significantly associated with the Likert scale. Furthermore, they created a word cloud to provide additional information into the teacher's performance, with positive remarks colored green and negative words colored red. High frequency terms were also displayed in a higher font size.

To analyze student textual comments, the Faculty Evaluation System [FES] was proposed [6]. They classified the system into three subcomponents, which are as follows: (Feature Extractor, sentiment analyzer, feature sentiment evaluator). In terms of stated accuracy, their findings are pretty promising. Another study [7] used SMS texts to assess classroom effectiveness. They began by identifying the many categories discussed in the SMS text, and then used three models to model the text: the base model, the corrected model, and the sentiment model. They came to the conclusion that the sentiment model can be utilized to evaluate teaching. A research work [8] presented a tool for evaluating teacher performance. Based on the combined score of opinion phrases, they classed the English and Filipino language comments as favorable or unfavorable. The scores were obtained from a polarity data set that they had produced. In required to conduct sentiment classification, they implement the Naive Bayes classifier. On an online database, various machine learning models (SVM, Random Forest, Logistics Regression, Decision Tree) and a deep learning method (multilayer perceptron) were utilized to do sentiment analysis [9]. The maximum accuracy was attained with SVM at 78.7% and Multilayer Perceptron at 78.3%.

Here, we used the historical data of online education for the assessment of students and teachers score evaluation base on different features. 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. For the analysis of different features, the proposed study will also train different machine learning model and evaluate the performance of different features set.

Chapter 2: Dataset Analysis

Dataset Overview

Introduction:

The dataset was prepared for the performance prediction of teachers and students by using python script. The students and teacher's dataset were divided into four categories including the Bad, Average, Good and Excellent category of students and teachers. The python script generated the 3000 samples in students and teacher's dataset for each category. The teacher's dataset was based on four input variables with one target variable while the student dataset was based on 13 input variables with one target variable. The generated datasets were chosen for the performance prediction of teachers and students in the proposed project.

The original Dataset contains the following number of rows and Columns:

Table 1: Counts of rows and columns in both datasets.

Dataset	Description	Count
Teachers Dataset	**Number of Rows in Dataset:**	12000
	Number of Variables in Dataset:	5
Students Dataset	**Number of Rows in Dataset:**	12000
	Number of Variables in Dataset:	14

Overview of Attributes

The python script generated the random value in specific range for teachers and students' attributes. For instance, the python script generated the 3000 samples of excellent students and for each samples script generated the random value of specific range for student-online-time, attendance, students-questions, and teacher-score. The script also generates the attribute values for each category in specific range. The snapshot of generating samples for the class of excellent teachers is shown in Figure 1:

```python
for i in range(0,3000):
    Student_Feed_Back.append('Excellent')
    Student_Online_time.append(random.randint(50, 60))
    Student_Attendance.append(random.randint(251, 365))
    Student_Questions.append(random.randint(80, 100))
    Teacher_Score.append(random.randint(80,100))
```

Figure 1: Code snippiest for generating samples of teacher's dataset.

The both datasets were saved in excel file and the variable detail of both datasets is available in the following table. The table described the variable name, description of variable, type of variable and range of the variable (Table 2).

Table 2: Overview of both dataset columns with type and range.

Teachers Dataset				
	Variable Name	**Description**	**Type**	**Range**
1	Feedback	Feedback of student for teachers in the form of category	Categorical	(Bad, Average, Good, Excellent)
2	Student-Online-Time	Time period of student in which he was online during the class in minutes.	Numeric	Bad (10, 30) Average (31, 40) Good (41, 49) Excellent (50, 60)
3	Students-Questions	How many questions was asked by the student?	Numeric	Bad (10, 30) Average (31, 49) Good (50, 79) Excellent (80, 100)
4	Student Attendance	Number of classes that student attend during the whole course.	Numeric	Bad (50, 150) Average (151, 180) Good (181, 250) Excellent (251, 365)
5	Teachers Score	The final performance score of the teachers.	Numeric	Bad (10, 30) Average (31, 49) Good (50, 79) Excellent (80, 100)
Students Dataset				
	Variable Name	**Description**	**Type**	**Range**
1	Teacher-Remarks	Feedback of teacher about the performance of the student.	Categorical	(Bad, Average, Good, Excellent)

2	Attendance-In-Course	Number of attended classes that student attend during course.	Numeric	Bad (50, 99) Average (100, 149) Good (150, 250) Excellent (251, 365)
3	Course-Access	Ho many times student access the course material on portal.	Numeric	Bad (5, 10) Average (11, 19) Good (20, 29) Excellent (30, 40)
4	Resource-Visit	How many times student access the provided resources for course learning.	Numeric	Bad (500, 999) Avg (1000,2999) Good (3000, 4999) Excellent (5000, 6000)
5	On-Time-Submission	How many times student submit the assignments of time?	Numeric	Bad (20, 39) Average (40, 59) Good (60, 79) Excellent (80, 100)
6	Exam1	Marks of student in Exam 1.	Numeric	Bad (20, 39) Average (40, 59) Good (60, 79) Excellent (80, 100)
7	Exam2	Marks of student in Exam 1.	Numeric	Bad (20, 39) Average (40, 59) Good (60, 79) Excellent (80, 100)
8	Exam3	Marks of student in Exam 1.	Numeric	Bad (20, 39) Average (40, 59) Good (60, 79) Excellent (80, 100)
9	Assignment 1	Score of students in Assignment 1.	Numeric	(30, 60)
10	Assignment 2	Score of students in Assignment 2.	Numeric	(50, 70)
11	Assignment 3	Score of students in Assignment 3.	Numeric	(60, 100)
13	Project	Marks of student in final project.	Numeric	Bad (20, 39) Average (40, 59) Good (60, 79) Excellent (80, 100)
14	Student-Score	Student performance/ Evaluation score.	Numeric	Bad (20, 39) Average (40, 59) Good (60, 79) Excellent (80, 100)

Cleaning and Preprocessing of Data

Description and Statistics of Dataset Attributes

The teachers and students' performance datasets have the 5 and 14 attributes respectively. The datatype of each attribute with the elementary statistics is following:

Table 3: General Statistics of each attribute in both datasets.

Teachers Dataset			
	Variable Name	**Type**	**Range**

1	Feedback	Categorical	`count 12000`
			`unique 4`
			`top Excellent`
			`freq 3000`
			`Name: Students feedback for teacher, dtype: object`

2	Student-Online-Time	Numeric	`count 12000.000000`
			`mean 38.881083`
			`std 13.495133`
			`min 10.000000`
			`25% 30.750000`
			`50% 40.500000`
			`75% 49.250000`
			`max 60.000000`
			`Name: Students Online Time, dtype: float64`

3	Students-Questions	Numeric	`count 12000.000000`
			`mean 53.596917`
			`std 27.008262`
			`min 10.000000`
			`25% 30.750000`
			`50% 49.500000`
			`75% 79.250000`
			`max 100.000000`
			`Name: Students Questions, dtype: float64`

4	Student Attendance	Numeric	`count 12000.000000`
			`mean 197.107667`
			`std 80.194802`
			`min 50.000000`
			`25% 150.750000`
			`50% 180.500000`
			`75% 250.250000`
			`max 365.000000`
			`Name: Students Attendance, dtype: float64`

5	Teachers Score	Numeric	`count 12000.000000` `mean 53.660000`

			std 27.124206 min 10.000000 25% 30.750000 50% 49.500000 75% 79.250000 max 100.000000 Name: Teachers Score, dtype: floa t64
Students Dataset			
	Variable Name	**Type**	**Range**
1	Teacher-Remarks	Categorical	count 12000 unique 4 top Good freq 3000 Name: Teacher Remarks, dtype: obj ect
2	Attendance-In-Course	Numeric	count 12000.000000 mean 176.509917 std 90.789687 min 50.000000 25% 99.750000 50% 149.500000 75% 250.250000 max 365.000000 Name: Attendance in course, dtype : float64
3	Course-Access	Numeric	count 12000.000000 mean 20.469500 std 10.648005 min 5.000000 25% 10.750000 50% 19.500000 75% 29.250000 max 40.000000 Name: Course Access, dtype: float 64
4	Resource-Visit	Numeric	count 12000.000000 mean 3062.168917 std 1874.974174 min 500.000000 25% 999.750000 50% 3000.000000 75% 4999.250000 max 6000.000000 Name: Resource Visit, dtype: floa t64
5	On-Time-Submission	Numeric	count 12000.000000 mean 59.650833 std 23.274315 min 20.000000 25% 39.750000 50% 59.500000

			75% 79.250000 max 100.000000 Name: On Time Submission, dtype: float64
6	Exam1	Numeric	count 12000.000000 mean 59.575417 std 23.284480 min 20.000000 25% 39.750000 50% 59.500000 75% 79.250000 max 100.000000 Name: Exam # 1, dtype: float64
7	Exam2	Numeric	count 12000.000000 mean 59.639250 std 23.336163 min 20.000000 25% 39.750000 50% 59.500000 75% 79.250000 max 100.000000 Name: Exam # 2, dtype: float64
8	Exam3	Numeric	count 12000.000000 mean 59.611333 std 23.262273 min 20.000000 25% 39.750000 50% 59.500000 75% 79.250000 max 100.000000 Name: Exam # 3, dtype: float64
9	Assignment 1	Numeric	count 12000.000000 mean 44.860583 std 8.919946 min 30.000000 25% 37.000000 50% 45.000000 75% 53.000000 max 60.000000 Name: Assignment # 1, dtype: float64
10	Assignment 2	Numeric	count 12000.000000 mean 59.940917 std 6.076115 min 50.000000 25% 55.000000 50% 60.000000 75% 65.000000 max 70.000000 Name: Assignment # 2, dtype: float64
11	Assignment 3	Numeric	count 12000.000000 mean 80.154833 std 11.882156

			min 60.000000 25% 70.000000 50% 80.000000 75% 91.000000 max 100.000000 Name: Assignment # 3, dtype: floa t64
13	Project	Numeric	count 12000.000000 mean 59.652750 std 23.270216 min 20.000000 25% 39.750000 50% 59.500000 75% 79.250000 max 100.000000 Name: Project, dtype: float64
14	Student-Score	Numeric	count 12000.000000 mean 59.687250 std 23.347595 min 20.000000 25% 39.750000 50% 59.500000 75% 79.250000 max 100.000000 Name: Student Scores, dtype: floa t64

Selection of Target Variable

The attribute "Teacher-Score" and "Student-Score" were set as target variable in teachers and students' datasets respectively. The possible range of teachers-score is 10 to 100 while the possible range of students-score was 20 to 100. The target variable was used for the prediction of teachers and students' performance based on the other attributes in both datasets.

As the target variable in both dataset is in numerical form that did not require the label encoding. Label encoding is a technique of converting string categorical value into numerical value.

Remove Missing Values and Outliers

Missing value in the tabular data usually contain the NAN, undefined and None values. As the datasets were generated by using the python script, so there is no chance for the presence of missing values. Both datasets were generated with controlled range of values for each attribute that remove the threat of missing values. The outliers are those sample that does not correlate with the other samples of same class. Due to the controlled values in attributes, removed the probability of happening outliers in both datasets.

Data Exploration

Explore Correlation of all attributes in Datasets

The strength of correlation can be characterized as strong (± 0.50 and ± 1). If the correlation value of two attributes exists between 0.50 and 1 then they are positively correlated. In the positively correlated attribute, the value of one attribute increases the value of other attribute increases or if the value of first variable decreases, then the value of other variable decreases.

If the correlation value of two attributes exists between -0.50 and -1 then they are negatively correlated. In the negatively correlated attribute, the value of one attribute increases the value of other attribute decreases or if the value of first variable decreases, then the value of other variable increases.

This section will check the correlations between all the variables within the dataset. From the analysis below we can conclude that the majority of variables in teacher's dataset show significant correlation. While from the student's dataset analysis, we can conclude that most of the variables are strongly correlated with the target variable except the Assigment1, Assignment 2 and Assignment 3 variable that have weak correlation.

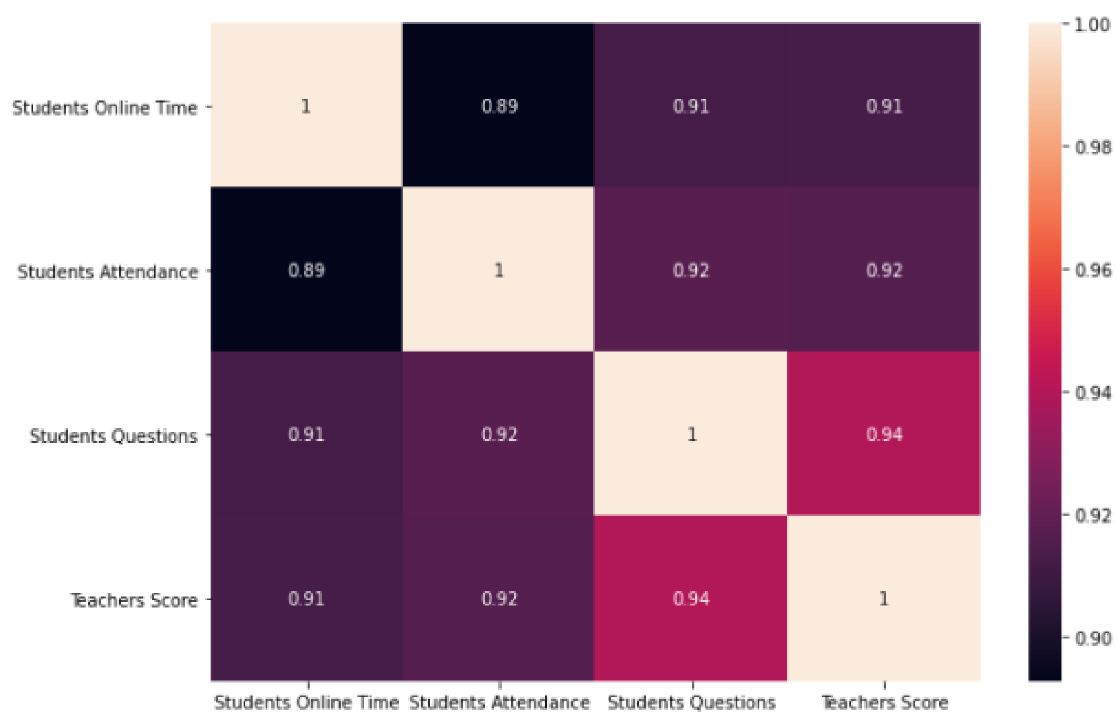


Figure 2: The correlation analysis of all variables in teacher's dataset.

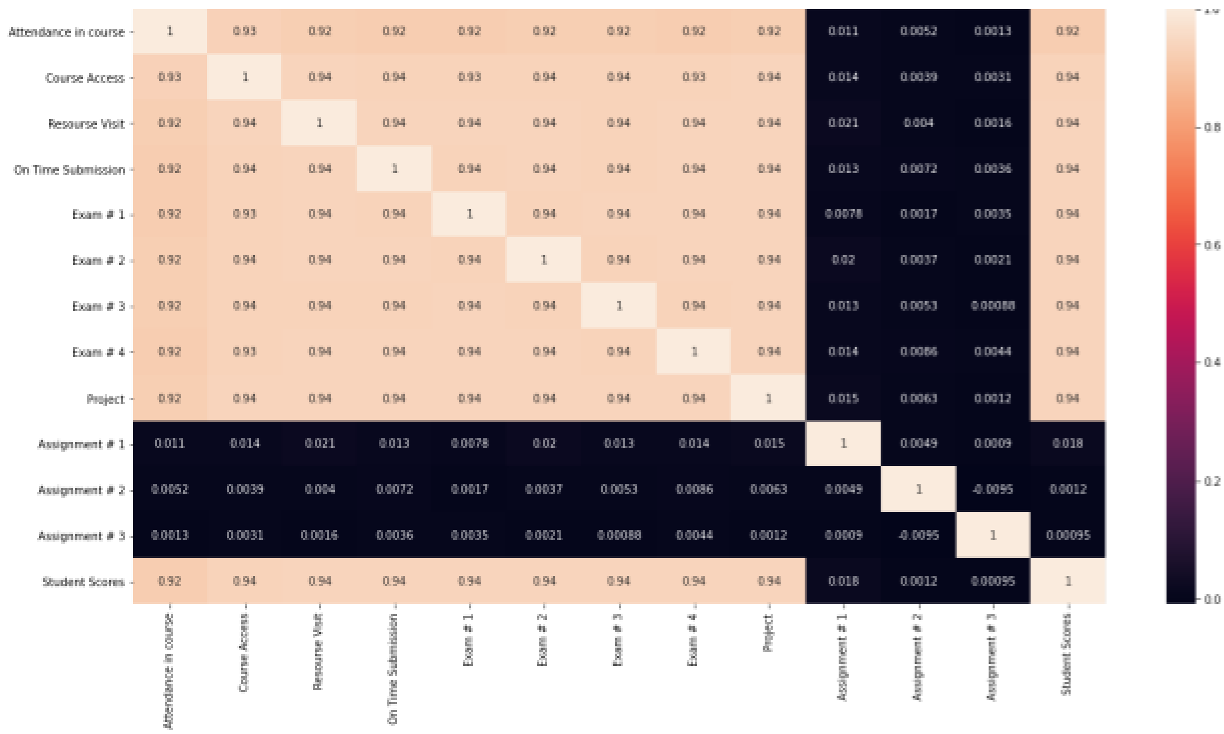


Figure 3: The correlation analysis of all variables in student's dataset.

Future Suggestions

The evaluations datasets presented in this project, can be used for predicting the variables "Teacher Score" and "Student Score". This column can be set as the target variable and all other variables can be used as input (predicting) variables. From the above data analysis, it can be concluded that regression and correlation analysis is the least promising approach. Another, possible and probably more promising approach will be to use a supervised machine learning algorithm that will predict the target variables into a decision tree. Since the target variable is numerical, a regressor tree can be used to determine the rules that will lead to the identification of the most promising decision path.

Quality of Data

The final set of data contains 5 and 13 columns for teachers and student's dataset. The total numbers of observations (rows) remaining in each dataset is 12000. Columns have been checked for the existence of extreme values (outliers) and actions have been taken to remove these observations out of the dataset. This decision has not been taken lightly. While outliers may indicate something scientifically interesting; in this scenario – where the objective is to simply predict a categorical variable using a classification decision tree; it is believed the presence of outliers may hinder the performance of the classifiers.

Furthermore, columns have also been checked for missing values. In particular the removal of missing values started by taking into consideration priority columns such as the target variable column precipitation type. Since the absence of a value within that column will render this column as impractical a decision has been taken to remove these observations out of the dataset. Additionally, columns have been checked for content validity by removing observations where incorrect data has been identified.

Furthermore, the original dataset has been split into two different datasets. One set will be used for training the data and the other will be used for validation purposes. In general, the remaining dataset has been greatly improved over its original version and it is ready for further manipulation.

Chapter 3:

Implementation

Machine Learning Model Implementation

After the cleaning and preprocessing of both datasets, identify the machine learning problem for target variable. As the target variable (Teacher-Score and Student-Score) in both datasets have the continuous value of score that indicate that the prediction of target variable made possible by regression models. Hence, different machine learning algorithms for regression analysis were chosen for the score prediction of students and teachers.

The list of all chosen regression models is below:

- Linear Regressor
- Random Forest Regressor
- KNN Regressor
- Decision Tree Regressor

After the training of the models, all models were tested on test set of the dataset to evaluate the performance of trained models. Below are some evaluation measures that were used to evaluate the performance of the model.

R2 Score - The R2 score is a very important metric that is used to evaluate the performance of a regression-based machine learning model. It is pronounced as R squared and is also known as the coefficient of determination. It works by measuring the amount of variance in the predictions explained by the dataset.

Mean Squared Error (MSE) – The Mean Squared Error (MSE) is perhaps the simplest and most common loss function, often taught in introductory Machine Learning courses. To calculate the MSE, you take the difference between your model's predictions and the actual value, square it, and average it out across the whole dataset.

The MSE will never be negative, since we are always squaring the errors. The MSE is formally defined by the following equation:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Root Mean Squared Error (RMSE) - RMSE is the standard deviation of the errors which occur when a prediction is made on a dataset. This is the same as MSE (Mean Squared Error) but the root of the value is considered while determining the accuracy of the model.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

Mean Absolute Error (MAE) - The Mean Absolute Error (MAE) is only slightly different in definition from the MSE, but interestingly provides almost exactly opposite properties! To calculate the MAE, you take the difference between your model's predictions and the actual value, apply the absolute value to that difference, and then average it out across the whole dataset.

The MAE, like the MSE, will never be negative since in this case we are always taking the absolute value of the errors. The MAE is formally defined by the following equation:

$$\text{MAE} = \frac{1}{n} \sum_{j=1}^{n} |y_j - \hat{y}_j|$$

Train Test Split of Dataset

The dataset of teachers and students' performance was split into training and testing set with the ratio of 70% and 30% respectively. The train test split function of Sklearn library was used to split the both datasets into training and testing set. The code sippet of splitting data into training and testing set is following:

```
from sklearn.model_selection import train_test_split

def split_data(data):
    train, test = train_test_split(data, test_size=0.33, random_state=42)
    return train, test
```

Figure 4: Code snippiest of Sklearn function for split training and testing data.

Linear Regressor

Linear Regressor is a machine learning model that used the state line to distinguished the different classes unlike the logistic regressor that used curved lines.

Objectives of the Model – The logistic Regressor is used in this project to predict the target variable for unseen data. In this project, Teachers-Score and Students-Score were used as target variable while the all-other attribute were used as input variable that shown in Dataset section.

Linear Regressor Analysis - The dataset was divided into two set for training, and testing with the ratio of 80% and 20% respectively. After the division of train test split, the training set contain the 8040 samples for both teachers and students' datasets. The training set was used for the training of the logistic regressor model of machine learning. Logistic Regressor was used with the default hyperparameters values for the training of both datasets. The code snippet of Logistic Regressor model is below:

```
# Applying the Linear Regression
lr = LinearRegression()
lr.fit(X_train, y_train)
y_pred = lr.predict(X_test)
model_evaluation(y_test, y_pred)
```

Figure 5: Code Snippiest of Linear Regression Model for training and predictions.

Linear Regressor Results – Linear Regressor model showed the 0.9175% and 0.9302% R2 score for teachers and students testing data. All the samples in testing dataset were passed to the trained model for predictions and the prediction results were compare with the ground truth values to compute the R2 Score. The model also showed the significant results for MSE, RMSE, and MAE. The snippet of logistic regression result for evaluation measures is below:

Table 4: Results of Linear Regression model for teachers and student's dataset on training data.

Result of Linear Regression Model ================================= R2 Score: 0.9175549650795715 Root Mean Squarred error: 7.727267418832785 Mean Squarred error: 59.710661762154686 Mean absolute error: 6.362078116918701	Result of Linear Regression Model ================================= R2 Score: 0.930227405540357 Root Mean Squarred error: 6.193568565485699 Mean Squarred error: 38.360291575372585 Mean absolute error: 5.268767412012073
Teachers Dataset	Studensts Dataset

Random Forest Regressor

Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.

Objectives of the Model – The Random Forest Regression is used in this project to predict the target variable for unseen data on the basis of trees prediction. In this project, Teachers-Score and Students-Score were used as target variable while the all-other attribute were used as input variable that shown in Dataset section.

Random Forest Regressor Analysis - The dataset was divided into two set for training, and testing with the ratio of 80% and 20% respectively. After the division of train test split, the training set contain the 8040 samples for both teachers and students' datasets. The training set was used for the training of Random Forest Regression model of machine learning. Random Forest Regression was used with the default hyperparameters values for the training of both datasets. The code snippet of Logistic Regressor model is below:

```
# Applying the random forest Regressor
rfr = RandomForestRegressor()
rfr.fit(X_train, y_train)
y_pred = rfr.predict(X_test)
model_evaluation(y_test,  y_pred)
```

Figure 6: Code Snippiest of Random Forest Model for training and predictions.

Random Forest Regressor Results – Random Forest Regression model showed the 0.9263% and 0.9361% R2 score for teachers and students testing data. All the samples in testing dataset were passed to the trained model for predictions and the prediction results were compare with the ground truth values to compute the R2 Score. The model also showed the significant results for MSE, RMSE, and MAE. The snippet of logistic regression result for evaluation measures is below:

Table 5:Results of Random Forest model for teachers and student's dataset on training data.

Teachers Dataset	Studensts Dataset
Result of Random Forest Regression Model ================================= R2 Score: 0.9263709259329824 Root Mean Squarred error: 7.302446074076769 Mean Squarred error: 53.3257186647922 Mean absolute error: 6.0298995517950065	Result of Random Forest Regression Model ================================= R2 Score: 0.9363961185958207 Root Mean Squarred error: 5.913441750227471 Mean Squarred error: 34.96879333333333 Mean absolute error: 5.092252525252525

KNN Regressor

K nearest neighbors is a simple algorithm that stores all available cases and predict the numerical target based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970's as a non-parametric technique.

Objectives of the Model – Here the KNN regressor was used for the regression analysis of target variable as the Teacher-score and student-score in selected datasets. The Score variables were set as target variable for KNN while rest of the attributes were set as input variable for KNN.

KNN Regressor Analysis - The dataset was divided into two set for training, and testing with the ratio of 80% and 20% respectively. After the division of train test split, the training set contain the 8040 samples for both teachers and students' datasets. The training set was used for the training of KNN Regression model of machine learning. KNN Regressor was also used with the default hyperparameters values for the training of both datasets. The code snippet of KNN model is below:

```python
# Applying the KNN Regressor
knn = KNeighborsRegressor()
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
model_evaluation(y_test, y_pred)
```

Figure 7: Code Snippiest of KNN Regressor Model for training and predictions.

KNN Regressor Results – Random Forest Regression model showed the 0.9278 % and 0.9282% score for teachers and students testing data. All the samples in testing dataset were passed to the trained model for predictions and the prediction results were compare with the ground truth values to compute the R2 Score. The model also showed the significant results for MSE, RMSE, and MAE. The snippet of logistic regression result for evaluation measures is below:

Table 6: Results of KNN Regressor model for teachers and student's dataset on training data.

Result of KNN Regression Model ================================= R2 Score: 0.9278530920649745 Root Mean Squarred error: 7.228572654837373 Mean Squarred error: 52.25226262626262635 Mean absolute error: 5.953232323232323	Result of KNN Regression Model ================================= R2 Score: 0.9282004408732857 Root Mean Squarred error: 6.282889221504464 Mean Squarred error: 39.474696969696964 Mean absolute error: 5.29469696969697
Teachers Dataset	Studensts Dataset

Decision Tree Regressor

Decision Tree is one of the most commonly used, practical approaches for supervised learning. It can be used to solve both Regression and Classification tasks with the latter being put more into practical application. It is a tree-structured classifier with three types of nodes.

Objectives of the Model – The Decision Tree Regressor is used in this project to predict the target variable for unseen data on the basis of trees prediction. In this project, Teachers-Score and Students-Score were used as target variable while the all-other attribute were used as input variable that shown in Dataset section.

Decision Tree Regressor Analysis - The dataset was divided into two set for training, and testing with the ratio of 80% and 20% respectively. After the division of train test split, the training set contain the 8040 samples for both teachers and students' datasets. The training set was used for the training of Decision Tree Regressor model of machine learning. KNN Regressor was also used with the default hyperparameters values for the training of both datasets. The code snippet of Decision Tree model is below:

```
# Appling the Decision tree Regressor
dtr = DecisionTreeRegressor()
dtr.fit(X_train, y_train)
y_pred = dtr.predict(X_test)
model_evaluation(y_test, y_pred)
```

Figure 8: Code Snippiest of Decision Tree Model for training and predictions.

Decision Tree Regressor Results – Decision Tree Regression model showed the 0.8776% and 0.8729 R2 score for teachers and students testing data. All the samples in testing dataset were passed to the trained model for predictions and the prediction results were compare with the ground truth values to compute the R2 Score. The model also showed the significant results for MSE, RMSE, and MAE. The snippet of logistic regression result for evaluation measures is below:

Table 7: Results of Decision Tree model for teachers and student's dataset on training data.

Result of Decision Tree Regression Model ================================= R2 Score: 0.8776612804463727 Root Mean Squarred error: 9.4129478891027 Mean Squarred error: 88.60358796296298 Mean absolute error: 7.505345117845117	Result of Decision Tree Regression Model ================================= R2 Score: 0.8709696151771196 Root Mean Squarred error: 8.422567688338662 Mean Squarred error: 70.93964646464646 Mean absolute error: 6.884595959595959
Teachers Dataset	Studensts Dataset

All the trained models showed the significant results in term of evaluation measures. Although, the Random Forest Regressor and KNN regressor showed the highest results for R2 score. The R2 score of Random Forest Regressor and KNN regressor was 0.9263 and 0.9278 respectively. Both showed the approximately equal results and the evaluation measures showed that the trained models are robust enough to made prediction on real data.

Chapter 4: Validation

Validation

Test Set

After the split of both datasets, the training data was used for the training of the models (Chapter 2) while the testing data will be used here for the validation of trained models. The testing set have the 3960 number of samples in teacher and student dataset. The details of testing set are given in below table.

Table 8: Counts of rows and columns in both test set.

Dataset	Description	Count
Teachers Dataset	Number of Samples in testing set	3960
	Total attributes in testing set	6
	Target Variable in testing set	Teacher Score
Students Dataset	Number of Samples in testing set	3960
	Total attributes in testing set	14
	Target Variable in testing set	Student Score

Target Variable Description

The attribute "Teacher-Score" and "Student-Score" were set as target variable in teachers and students' test set respectively. The possible range of teachers-score is 10 to 100 while the possible range of students-score was 20 to 100. The target variable was used for the prediction of teachers and students' performance based on the other attributes in both datasets.

Evaluation Measures

For the validation of the models, we chose some evaluation measures to measure the performance of our trained model. Evaluation measures were selected based on the nature of proposed problem. As the evaluation score prediction of teachers and students is a regression problem, that's why we chose all the evaluation measures that are used for regression problems. To evaluate the performance of the trained model, we used the same evaluation measures that were used in training section of the model (R2, MSE, RMSE, MAE).

Validation

All models were tested on test set of the dataset to evaluate the performance of trained models. The input variables of students and teachers test set were passed to each model and make prediction on each sample. Every model returns a list of predations base on its learning equal to the length of testing examples (rows). The predicted scores were labeled as y’ while the actual scores were labeled as y. The score of all evaluation measures were calculated with y (actual scores) and y’ (predicted scores) by using the relative formulas (described above).

Linear Regressor Validation: The input variables of all the testing examples were passed to the Linear Regressor model and it return the predicted scores list (y’) equal to the length of testing examples (3960). The actual scores/target variables (y) and predicted scores (y’) were passed to the evaluation measure function that calculate the R2 score, MSE, RMSE, MAE by using the equation 1-4 respectively. Linear Regressor Model showed the 0.917555, 7.727267, 59.71066, and 6.362078 values of R2 Score, RMSE, MSE, MAE for teachers’ dataset respectively. The Linear Regression model also showed the 0.930227, 6.193569, 38.36029, and 5.268767 values of R2 Score, RMSE, MSE, MAE for students’ dataset respectively.

Random Forest Regressor Validation: The input variables of all the testing examples were passed to the Random Forest Regressor model and it return the predicted scores list (y’) equal to the length of testing examples (3960). The actual scores/target variables (y) and predicted scores (y’) were passed to the evaluation measure function that calculate the R2 score, MSE, RMSE, MAE by using the equation 1-4 respectively. Random Forest Regressor Model showed the 0.92693, 7.274672, 52.92085, and 6.0097 values of R2 Score, RMSE, MSE, MAE for teachers’ dataset respectively. The Random Forest Regression model also showed the 0.936384, 5.913984, 34.97521, and 5.094838 values of R2 Score, RMSE, MSE, MAE for students’ dataset respectively.

K-Nearest Neighbor Validation: The input variables of all the testing examples were passed to the K-Nearest Neighbor model and it return the predicted scores list (y’) equal to the length of testing examples (3960). The actual scores/target variables (y) and predicted scores (y’) were passed to the evaluation measure function that calculate the R2 score, MSE, RMSE, MAE by using the equation 1-4 respectively. K-Nearest Neighbor Model showed the 0.928033, 7.219567, 52.12214, and 5.948182 values of R2 Score, RMSE, MSE, MAE for teachers’ dataset respectively. The K-Nearest Neighbor model also showed the 0.928191, 6.283286, 39.47969, and 5.295505 values of R2 Score, RMSE, MSE, MAE for students’ dataset respectively.

Decision tree Regressor Validation: The input variables of all the testing examples were passed to the Decision tree Regressor model and it return the predicted scores list (y') equal to the length of testing examples (3960). The actual scores/target variables (y) and predicted scores (y') were passed to the evaluation measure function that calculate the R2 score, MSE, RMSE, MAE by using the equation 1-4 respectively. Decision tree Regressor Model showed the 0.877154, 9.432427, 88.97068, and 705246 values of R2 Score, RMSE, MSE, MAE for teachers' dataset respectively. The Decision tree Regression model also showed the 0.872907, 8.359081, 69.87424and 6.832828 values of R2 Score, RMSE, MSE, MAE for students' dataset respectively.

Validation Results

After the validation of all the models using test set of teachers and student data, the models were compared base on the evaluation scores. Random Forest Regressor Model Showed the 0.92693 and 0.936384 highest R2 Score for teachers and student test data respectively. The Random Forest model also showed the lowest error for teachers and student test data. The comparative table of evaluation measures for teacher and student test set is available below:

Table 9: 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 10: 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

The below bar plot is also showed that Random Forest Regressor showed the best R2 Score for teachers and students' dataset.

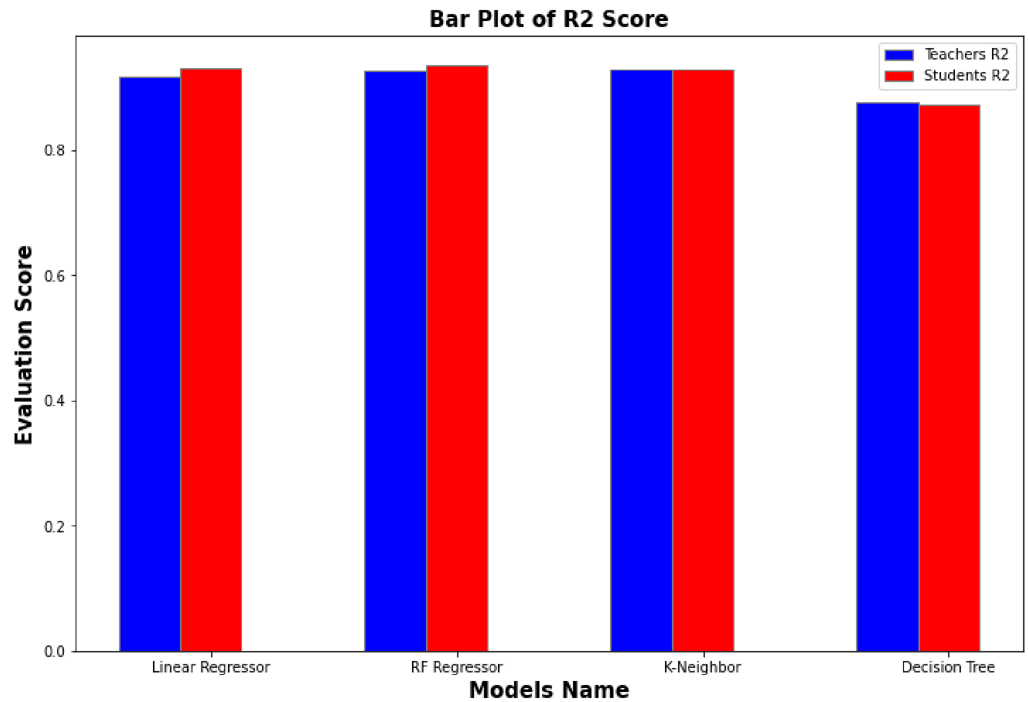


Figure 9: Comparison of different models using RMSE and MAE in bar plot.

The below bar plot is also showed that Random Forest Regressor showed the lowest RMSE and MAE error Score for teachers and students' dataset.

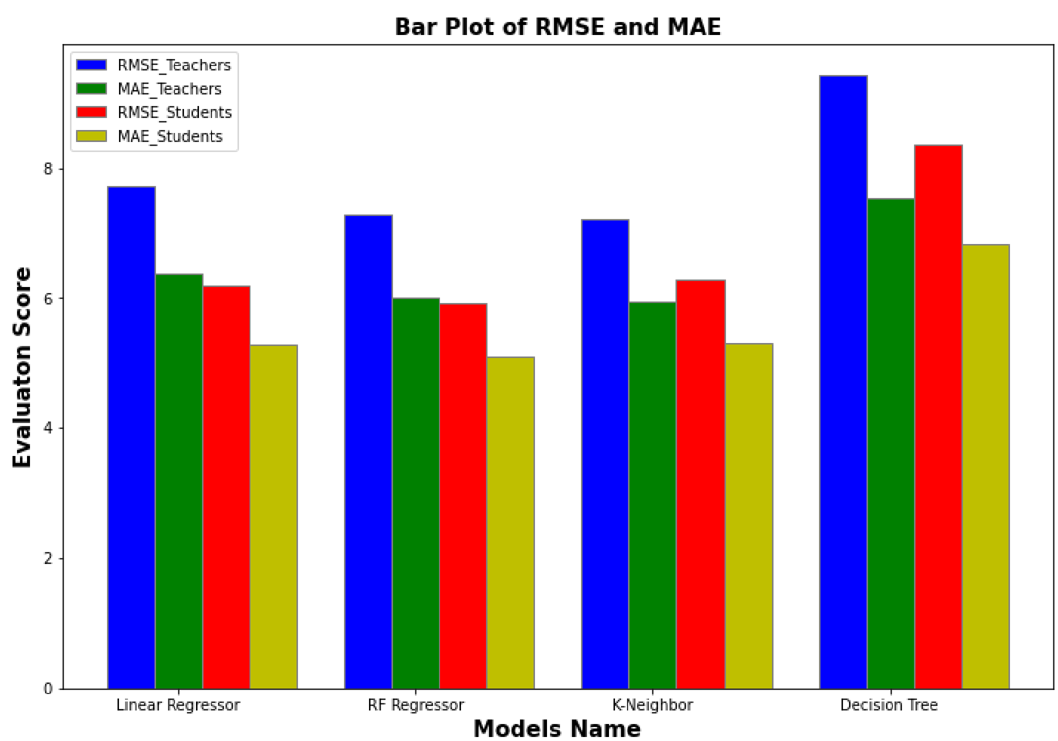


Figure 10: Comparison of different models using RMSE and MAE in bar plot.

By the study of comparative table of evaluation measures and bar plot of evaluation measures, it is clear that Random Forest Regressor outer perform by all models in term of R2 Score and other evaluation measures. The best result conclude that Random Forest is the best model for deployment in real world environment to make predictions on real world data.

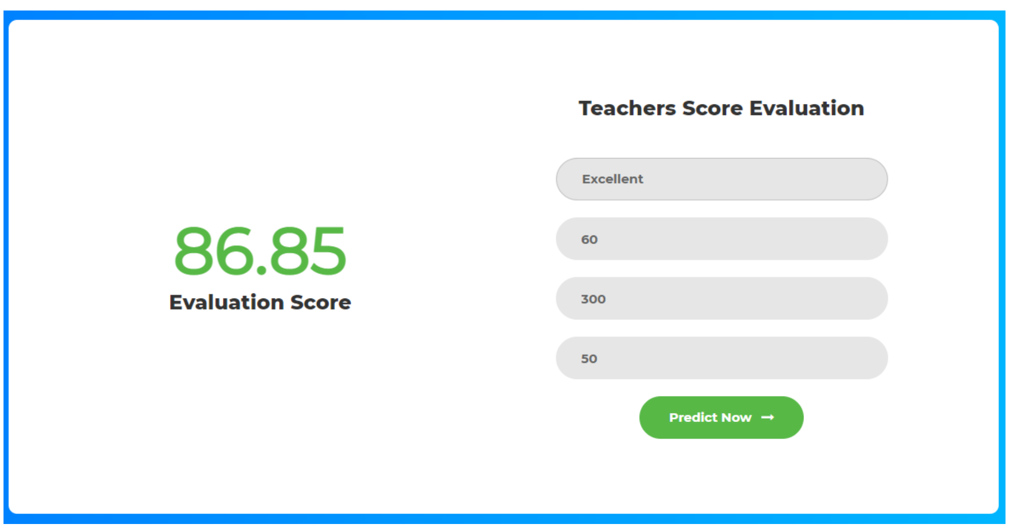
Chapter 4: Deployment

Deployment

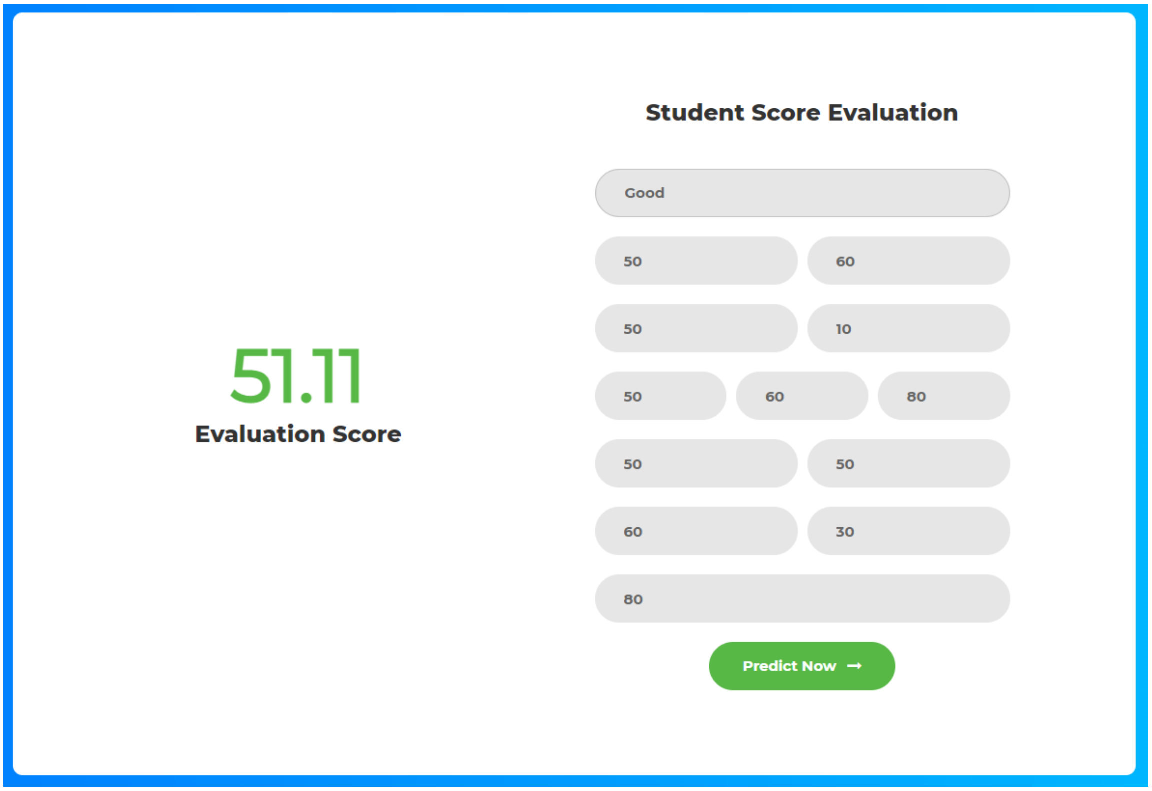
After the validation of all models, results conclude theta Random Forest Regressor is the best model for the evaluation score prediction of teachers and students. We made an interface for getting the data from user and made prediction on the collected data by Random Forest Regression. The Deployment of the best model is based on two phases: front-end phase and backend phase. The detail working of each phase is described below:

Frontend Phase

In the frontend phase, we developed the 2 screens for getting the teachers and students input variable data respectively. The teacher interface screen is based on the 5 input fields followed by the submit button while the student interface was based on the 14 input fields based on the input variables of student dataset followed by the submit button. In the teachers Interface Screen, one input field is the dropdown filed that take categorical values from dropdown while rest of the fields are numerical fields that take integral values. In the last of the screen a submit button is place to submit the user entered data at server side. The overview of teacher's screen is below:



In the student interface screen, one field is dropdown field that take categorical values from user while rest of the fields take the numerical values. There is also a submit button to send the user entered value at the server side. The snippet of Student Screen is presented below:



Backend Phase

The backend phase base on several functions with the two main functions that calculate the teachers and students' evaluation score. Both functions received the form data from user interface via hyper parameters. After converting the categorical data into numerical representation using one hot encoding, both functions send the converted to Random Forest Regressor model that calculate the evaluation encode bae of the input variables. After the calculation of evaluation score, both functions return the score towards the frontend interface in JSON format. The interface is capable to receive the score and show to the user as show in above figures.

Configurations

The frontend and backend phases were developed by using the Django Framework. The Django Framework of Python is the secure and stable framework for the development of user friendly, secure and attractive web application. The MVC (Model, view, Controller) architecture was followed to developed the whole application. After the complete development of application, the application was deployed on the local server with 127.0.0.1 IP address. Minimum 4GB Ram and 500GB are required to run the application on local server.

After the complete deployment of application, the application was tested by entering the values in the form and calculated the teachers and students' evaluation scores. After, clicking of the submit button, proposed model, and developed application perform well and showed the evaluation scores for both teachers and students also as shown in above figures.

Chapter 5:

Discussion & Future Directions

Discussion

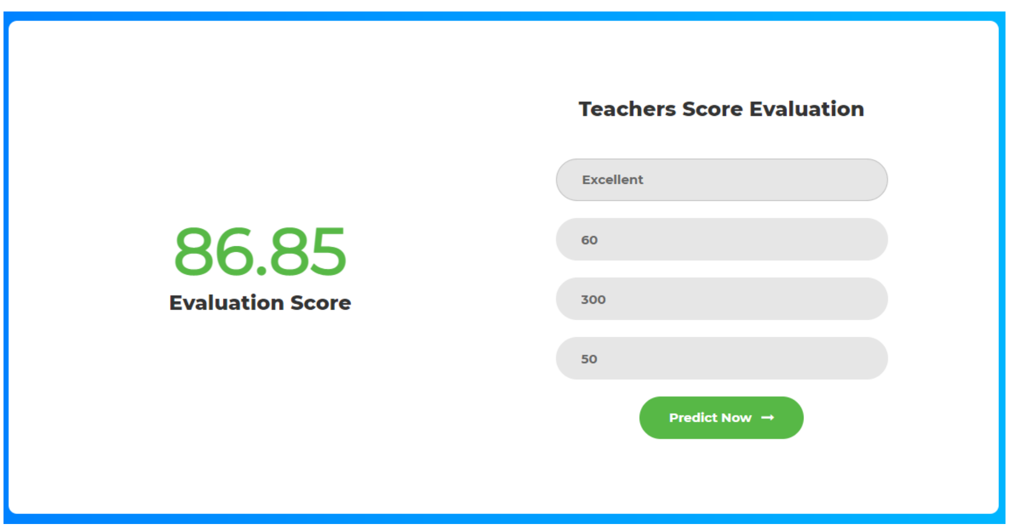
In the proposed study, we trained the different machine learning models for different features of the students and teachers. The identification of most correlated features for the better MSE of the machine learning model was a biggest challenge. Sometimes, teachers are unnecessarily set the paper or 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.

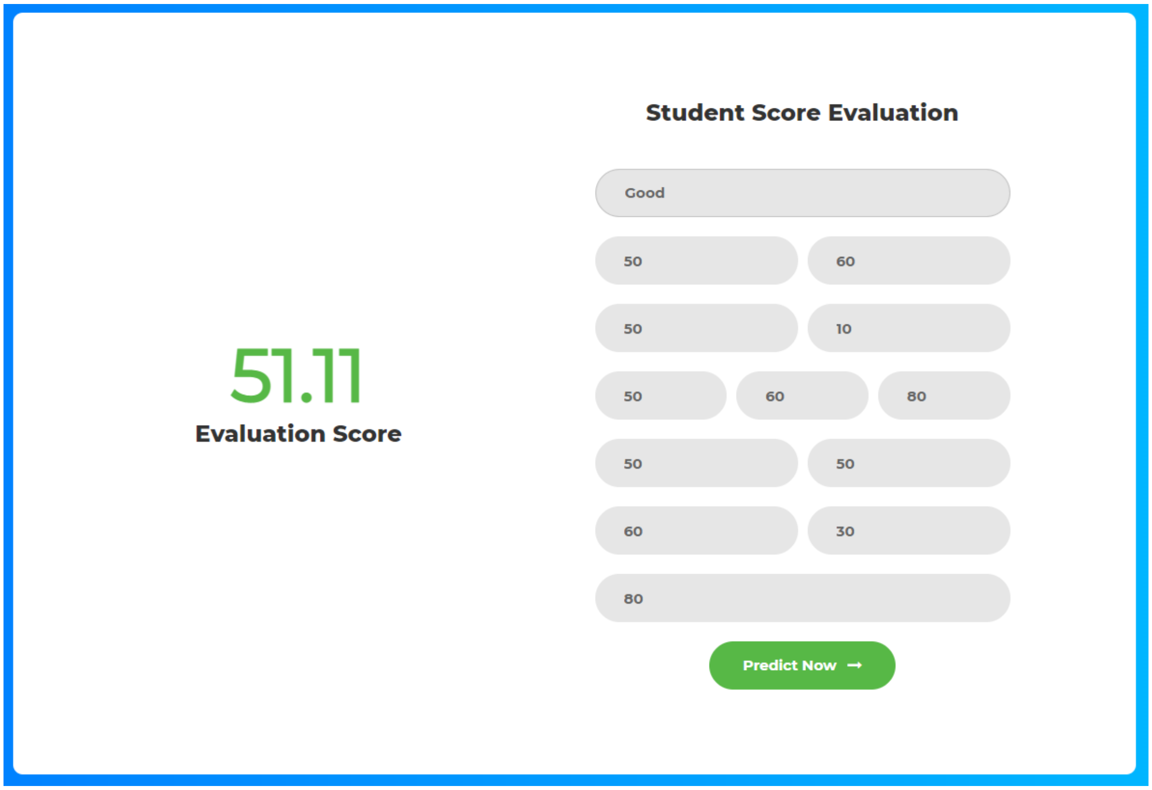
Here, we generated our own historical data for the teacher and student final grade prediction. The proposed study contributes in the final assessment of the student score and teacher score for online offered courses. Although, we use correlation calculation technique for the selection of features, but 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.

Future Directions and Scope

In this project, we trained the different machine learning for the performance assessment of teachers and student in online teaching mode. The proposed results showed the result that are robust enough for the assessment of teachers and students in online teaching model. In online teaching mode, teachers and student related data like online teaching time of teachers and duration of attending class by student can be collected easily. The rest of the performance data like assignment marks and exam marks of students can be collected from the LMS of concerned organization department. After compiling the data of students and teachers, the evaluation score of each teacher and student can be calculated one by one as show in below figures.

The automatic calculation of finding the evaluation score for teachers and students will reduce the effort of generating complex criteria of calculating score for teachers and students. The proposed project will also reduce the human effort to calculate the score manually for each teacher and student by selected criteria. Most prominently, the teacher's evaluation usually calculated by taking the feedback from students and the totally score of teachers is depended on the feedback of students. This calculated score usually don't cater the personal interest of students in bad and good feedback. The proposed project also contributing the other measures to calculate the teachers and students score that can be collected automatically and reduce. Hence the proposed project will reduce the dependency of teacher and students' personal interests on giving tine of feedback and calculate the performance score totally based on performance measure.





References

[1] A. Ortigosa, J. M. Martín, and R. M. Carro, "Sentiment analysis in Facebook and its application to e-learning," *Computers in Human Behavior*, vol. 31, no. 1, pp. 527–541, Feb. 2014, doi: 10.1016/J.CHB.2013.05.024.

[2] C. Pong-Inwong and W. S. Rungworawut, "Teaching senti-lexicon for automated sentiment polarity definition in teaching evaluation," *Proceedings - 2014 10th International Conference on Semantics, Knowledge and Grids, SKG 2014*, pp. 84–91, Nov. 2014, doi: 10.1109/SKG.2014.25.

[3] L. C. Yu *et al.*, "Improving early prediction of academic failure using sentiment analysis on self-evaluated comments," *Journal of Computer Assisted Learning*, vol. 34, no. 4, pp. 358–365, Aug. 2018, doi: 10.1111/JCAL.12247.

[4] M. A. Ullah, "Sentiment analysis of students feedback: A study towards optimal tools," *IWCI 2016 - 2016 International Workshop on Computational Intelligence*, pp. 175–180, Feb. 2017, doi: 10.1109/IWCI.2016.7860361.

[5] Q. Rajput, S. Haider, and S. Ghani, "Lexicon-Based Sentiment Analysis of Teachers' Evaluation," *Applied Computational Intelligence and Soft Computing*, vol. 2016, pp. 1–12, 2016, doi: 10.1155/2016/2385429.

[6] A. Kumar and R. Jain, "Faculty Evaluation System," *Procedia Computer Science*, vol. 125, pp. 533–541, Jan. 2018, doi: 10.1016/J.PROCS.2017.12.069.

[7] C. K. Leong, Y. H. Lee, and W. K. Mak, "Mining sentiments in SMS texts for teaching evaluation," *Expert Systems with Applications*, vol. 39, no. 3, pp. 2584–2589, Feb. 2012, doi: 10.1016/J.ESWA.2011.08.113.

[8] F. F. Balahadia, M. C. G. Fernando, and I. C. Juanatas, "Teacher's performance evaluation tool using opinion mining with sentiment analysis," *Proceedings - 2016 IEEE Region 10 Symposium, TENSYMP 2016*, pp. 95–98, Jul. 2016, doi: 10.1109/TENCONSPRING.2016.7519384.

[9] J. Sultana, N. Sultana, K. Yadav, and F. Alfayez, "Prediction of Sentiment Analysis on Educational Data based on Deep Learning Approach," *21st Saudi Computer Society National Computer Conference, NCC 2018*, Dec. 2018, doi: 10.1109/NCG.2018.8593108.