DEPARTMENT OF COMPUTER & INFORMATION SYSTEMS ENGINEERING
BACHELORS IN COMPUTER SYSTEMS ENGINEERING
Course Code: CS-324
Course Title: Machine Learning
Complex Engineering Problem
TE Batch 2019, Spring Semester 2022
Grading Rubric
TERM PROJECT

Group Members:

Student No.	Name	Roll No.
S1	**Rimsha Sohail**	**CS-19051**
S2	**Sidra Fatima**	**CS-19117**
S3	**Sana Parveen**	**CS-19123**

CRITERIA AND SCALES				**Marks Obtained**		
				S1	**S2**	**S3**
Criterion 1: Does the application meet the desired specifications and produce the desired outputs? (CPA-1, CPA-2, CPA-3) [8 marks]						
1	2	3	4			
The application does not meet the desired specifications and is producing incorrect outputs.	The application partially meets the desired specifications and is producing incorrect or partially correct outputs.	The application meets the desired specifications but is producing incorrect or partially correct outputs.	The application meets all the desired specifications and is producing correct outputs.			
Criterion 2: How well is the code organization? **[2 marks]**						
1	2	3	4			
The code is poorly organized and very difficult to read.	The code is readable only to someone who knows what it is supposed to be doing.	Some part of the code is well organized, while some part is difficult to follow.	The code is well organized and very easy to follow.			
Criterion 3: Does the report adhere to the given format and requirements? **[6 marks]**						
1	2	3	4			
The report does not contain the required information and is formatted poorly.	The report contains the required information only partially but is formatted well.	The report contains all the required information but is formatted poorly.	The report contains all the required information and completely adheres to the given format.			
Criterion 4: How does the student performed individually and as a team member? (CPA-1, CPA-2, CPA-3) **[4 marks]**						
1	2	3	4			
The student did not work on the assigned task.	**The student worked on the assigned task, and accomplished goals partially.**	**The student worked on the assigned task, and accomplished goals satisfactorily.**	**The student worked on the assigned task, and accomplished goals beyond expectations.**			

Final Score = (Criterial_1_score x 2) + (Criteria_2_score / 2) + (Criteria_3_score x (3/2)) + (Criteria_4_score)
= _____

Introduction:

We have develop three different Machine learning models, on each model two different algorithms: Multiple Linear Regression and Random Forest Regression have been applied. These models are used to predict final CGPA of a student at the end of fourth year given GPs of the courses obtained in initial years (up to first, second or third year). The output is shown using interface.

Data Preprocessing:

Data preprocessing in Machine Learning is the first crucial step marking the initiation of the process that helps enhance the quality of data to promote the extraction of meaningful insights from the data.

Typically, real-world data is incomplete, inconsistent, inaccurate (contains errors or outliers), and often lacks specific attribute values. This is where data preprocessing comes, a data mining technique – it helps to clean, format, and organize the raw data, thereby making it ready-to-go for Machine Learning models.

Steps in Data Preprocessing in Machine Learning:

There are seven significant steps in data preprocessing in Machine Learning:

1. **Acquire the dataset:** Acquiring the dataset is the first step in data preprocessing, to build and develop Machine Learning models. For the project, the Prediction of Cumulative Grade Point Average, the dataset, "The_Grades_Dataset.csv" has been given.

2. **Importing all the crucial libraries:** The predefined Python libraries can perform specific data preprocessing jobs. Importing all the crucial libraries is the second step in data preprocessing in machine learning. Following Python libraries are imported to be used for this our project:
    - **NumPy**: When it comes to scientific computing, NumPy is one of the fundamental packages for Python providing support for large multidimensional arrays and matrices along with a collection of high-level mathematical functions to execute these functions swiftly.
    - **Pandas:** Pandas is an excellent open-source Python library for data manipulation and analysis. It is extensively used for importing and managing the datasets. It packs in high-performance, easy-to-use data structures and data analysis tools for Python.
    - **Matplotlib:** Matplotlib is a Python 2D plotting library that is used to plot any type of charts in Python.
    - **Scikit- learn:** It is a free software machine learning library for the Python programming language and can be effectively used for a variety of applications which include classification, regression, clustering, model selection, naive Bayes', grade boosting, K-means, and preprocessing.
    - **Gradio:** Gradio is an open-source python library that permits you to rapidly make simple to utilize, adjustable UI parts for the ML model, any API, or any subjective capacity in only a couple of lines of code. You can coordinate the GUI straightforwardly into your Python notebook, or you can share the link with anybody. Gradio helps in building an online GUI in a couple of lines of code which is convenient for showing exhibitions of the model presentation. It is quick, simple to set up, and prepared to utilized. Gradio works with a wide range of media-text, pictures, video, and sound.
    - **Seaborn:** Saeborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

3. **Import the dataset:** After acquiring the dataset, “The_Grades_Dataset.csv”, it is important to import that the dataset needed in the Jupyter Notebook environment where whole implementing machine learning systems’ process is being done. We uploaded the given dataset in the file area of Jupyter notebook and imported it inside the notebook through “read_csv()” function of the Pandas library, inside the variable, ‘dataset.’
   After importing the dataset, the unnecessary column, “Seat No.” has been dropped as it does not have any impact on the courses or CGPA values.

4. **Identifying and handling the missing values**: Data Cleaning is the process of finding and correcting the inaccurate/incorrect data that are present in the dataset. One such process needed is to do something about the values that are missing in the dataset. In data preprocessing, it is pivotal to identify and correctly handle the missing values, failing to do this, it might draw inaccurate and faulty conclusions and inferences from the data. In real life, many datasets will have many missing values, so dealing with them is an important step.

   The given dataset, “The_Grades_Dataset.csv”, has the missing values which we have found out through using “df.info()”, it gives non-null values counts in each column. The given dataset have 571 rows and 43 columns. For filling missing values, there are many methods available, but choosing the best method according to the type of missing values and its significance, is important before we start filling/deleting the data.

   There are many different methods that can be used to deal with the missing data. We have dealt with missing values by following 2 methods:
   1) **Deleting the rows that have more than 10 missing values:** Firstly, we have checked for total missing values count at each row in data frame, by using “iloc[i].isnull().sum()” inside a ‘for loop’ and after getting the no. of missing values in each row, we have dropped those rows that have more than 10 missing values by using “.drop” instruction.

   2) **Filling the remaining missing values with Mode:** We have filled the missing data with mode, using “df[this_column].fillna(df[this_column].mode()[0])” inside a ‘for loop’ because the given dataset is categorical in nature. Here the missing values are replaced with the mode value or most frequent value of the entire feature column. When the data is skewed, it is good to consider using mode values for replacing the missing values. Imputing missing data with mode values can be done with numerical and categorical data.

5. **Encoding the categorical data:** Categorical data refers to the information that has specific categories within the dataset. In the given dataset, the grades A+, A, A-, B+, B, B-, C+, C, C-, D+, D, F, I, W, WU are used which is categorical data.
   Machine Learning models are primarily based on mathematical equations thus we can understand that keeping the categorical data in the equation will cause certain issues since we would only need numbers in the equations. So categorical data must be converted into numerical values.

   To do so, we have used mapping technique to convert ordinal data into numerical values. Through data mapping technique, we have set values for each of the above mentioned grades.

   dataMapping = {"A+":4.0, "A":4.0, "A-":3.7, "B+":3.4, "B":3.0, "B-":2.7, "C+":2.4, "C":2.0, "C-":1.7, "D+":1.4, "D":1.0, "F":0.0, "I":0.0, "W":0.0, "WU":0.0}

By using a 'for loop', we have mapped the above techniques on all the columns except for CGPA column as it is already in numerical.

6. **Splitting the dataset**: Splitting the dataset is the next step in data preprocessing in machine learning. Every dataset for Machine Learning model must be split into two separate sets – training set and test set.
   Training set denotes the subset of a dataset that is used for training the machine learning model. Here, we are already aware of the output. A test set, on the other hand, is the subset of the dataset that is used for testing the machine learning model. The ML model uses the test set to predict outcomes.

   Firstly we have extract all the courses columns in 'X' variables and CGPA column in 'Y' variable through using ".iloc" instruction. Then we have split the dataset in training and testing set by importing train_test_split from sklearn.model_selection.

7. **Feature scaling:** Feature scaling marks the end of the data preprocessing in Machine Learning. It is a method to standardize the independent variables of a dataset within a specific range. In other words, feature scaling limits the range of variables so that you can compare them on common grounds. Standardization and Normalization method can be used for feature scaling. We have used the standardization method. To do so, we have import StandardScaler class of the sci-kit-learn library. We have created the object of StandardScaler class and then fit and transform the training and testing dataset.

After doing all the data preprocessing steps, we have implement Multiple Linear Regression algorithm on whole dataset to check for accuracy.

Models:

After implementing the data preprocessing steps and checking the accuracy of final dataset, we have developed three different models in which dataset's features (courses) that are inputs, have been chosen differently in each model however the target, the output, that is CGPA column is same in all three models. Two different algorithms have been applied on each model. We have developed following models:

- **Model 1: Predicting final CGPA based on GPs of first year only:** In this model, we have selected all the columns of courses of first years as input in variable, "datasetFeatures1" and have selected CGPA column as output in variable, "datasetTarget."
- **Model 2: Predicting final CGPA based on GPs of first two years:** In this model, we have selected all the columns of courses of first year and second year as input in variable, "datasetFeatures2" and have selected CGPA column as output in variable, "datasetTarget."
- **Model 3: Predicting final CGPA based on GPs of first three years: In** this model, we have selected all the columns of courses of first year, second year and third year as input in variable, "datasetFeatures3" and have selected CGPA column as output in variable, "datasetTarget."

Machine Learning Algorithms:

As our project is supervised learning task so we have chosen Multiple Linear Regression and Random Forest Regression algorithm. These algorithms are supervised learning algorithms. Supervised learning, also known as supervised machine learning, is a subcategory of machine learning and artificial

intelligence. It is defined by its use of labeled datasets to train algorithms that to classify data or predict outcomes accurately. These algorithms consists of a target/outcome variable (or dependent variable) which is to be predicted from a given set of predictors (independent variables). Using this set of variables, we generate a function that map inputs to desired outputs. The training process continues until the model achieves a desired level of accuracy on the training data.

Regression models are used to describe relationships between variables by fitting a line to the observed data. Regression allows you to estimate how a dependent variable changes as the independent variable(s) change.

- **Multiple Linear Regression Algorithm**: Multiple linear regression refers to a statistical technique that is used to predict the outcome of a variable based on the value of two or more variables. It is sometimes known simply as multiple regression, and it is an extension of linear regression. The goal of multiple linear regression is to model the linear relationship between the explanatory (independent) variables and response (dependent) variables.
- **Random Forest Regression Algorithm:** A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems. A random forest algorithm consists of many decision trees. The (random forest) algorithm establishes the outcome based on the predictions of the decision trees. It predicts by taking the average or mean of the output from various trees. Increasing the number of trees increases the precision of the outcome. A random forest eradicates the limitations of a decision tree algorithm. It reduces the overfitting of datasets and increases precision. It can produce a reasonable prediction without hyper-parameter tuning. It solves the issue of overfitting in decision trees.

Interfaces:

There are two interfaces implemented in our Jupyter Notebook.

- **Predicting final CGPA output:** This interface is regarding where the user can easily interact and select the model and then select which algorithm he/she want to implement and then user will be directed to gradio environment where GPAs of the required courses will be given and the predicted CGPA that is the output will be shown at the end.
- **Comparison of all the models:** This interface is regarding where the user will be directed to gradio environment where he/she has to input all GPAs of the first, second and third year courses and then in output all the predicted CGPA according to all the models along with their algorithms each will be shown at the end.

Distinguishing Features:

- **Graph for missing values:** We have added the graphs at two different places. One before dealing with missing values and one after missing values have been removed. When there are missing values, lines will be shown against the features which has missing values. When there are no missing values present in our dataset, the graph will be clean.

- **Comparison of all the models**: This interface is regarding where the user will be directed to gradio environment where he/she has to input all GPAs of the first, second and third year courses and then in output all the predicted CGPA according to all the models along with their algorithms each will be shown at the end.

Tabular Comparison of all the Models:

For machine learning model, it is necessary to obtain the accuracy on training data, but it is also important to get a genuine and approximate result on unseen data otherwise Model is of no use. So to build and deploy a generalized model we require to evaluate the model which helps us to better optimize the performance, fine-tune it, and obtain a better result. For this, we have used the following evaluation metric for our models.

R Squared (R2): R-Squared (R^2 or the coefficient of determination) is a statistical measure in a regression model that determines the proportion of variance in the dependent variable that can be explained by the independent variable. In other words, r-squared shows how well the data fit the regression model (the goodness of fit). Maximum value of R2 is 1. So, higher the value of R2, better the model is.

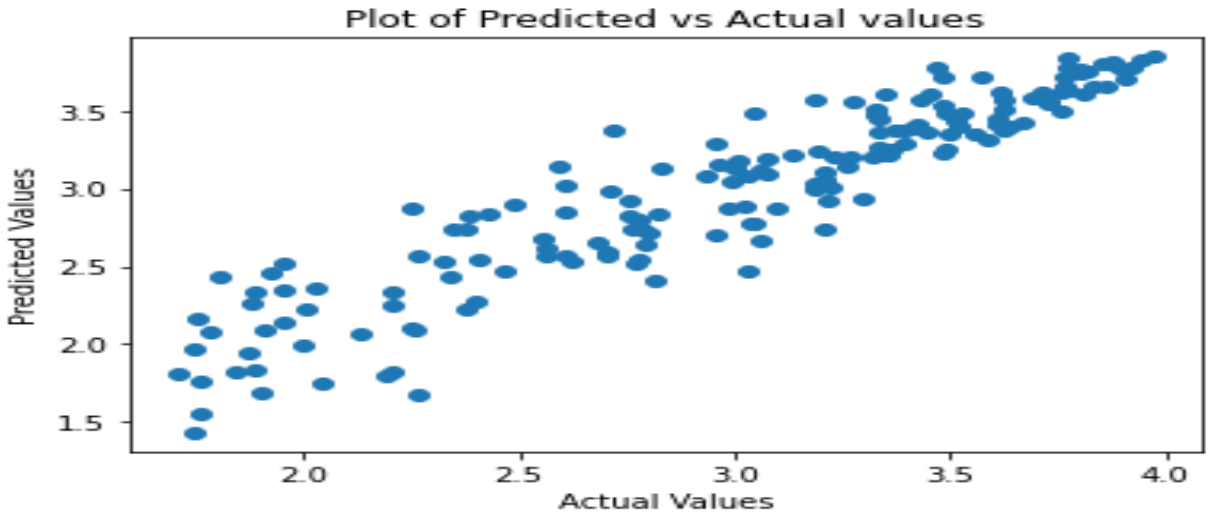
Models	R-Squared Value
Model 1 with Multiple Linear Regression Algorithm	0.8623875820753107
Model 1 with Random Forest Regression Algorithm	0.9743453507464857
Model 2 with Multiple Linear Regression Algorithm	0.9540068028905622
Model 2 with Random Forest Regression Algorithm	0.9857702591643975
Model 3 with Multiple Linear Regression Algorithm	0.9946197427939577
Model 3 with Random Forest Regression Algorithm	0.9924182659167852

Comments:

After implementing both the machine learning algorithms on all three models, it can be said that Model 3 with Random Forest Regression Algorithm's performance is better than other models. Random Forest Regression Algorithm has shown better performance than Multiple Linear Regression Algorithm in all three models.

The averaging makes a Random Forest better than a single Decision Tree hence improves its accuracy and reduces overfitting. A prediction from the Random Forest Regression algorithm is an average of the predictions produced by the trees in the forest.

After going through all the training and testing set accuracies of all different models, we can say that there is little under fitting in model 1 with multiple linear regression.



For other models, there is no underfitting or overfitting issue. All these models represent balanced fit.

For more improvement of the models, we should increase dataset values. Increasing the training set data will have more data to train on. Using some of the dataset portion as validation set to evaluate the model during the development stages and then finally using test set data for final evaluation of a model.