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	**Wajiha Khan**	**CS-19064**
S2	**Hafiza Javeria Adil**	**CS-19301**

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)
= _____

Project Description:

The objective of this project is to predict the CGPA of students. Cumulative Grade Point Average (CGPA) refers to overall Grade Point Average (GPA), obtained by dividing the total Grade Poiints (GPs) earned in all courses attempted by the total degree-credit hours in all attempted courses. CGPA is used for an assessment of a student's overall academic standing. There are three models in this project . Model 1 predicts CGPA based on GP's of first year only. Model 2 predicts CGPA based on GP's of first two years. Model 3 predicts CGPA based on GP's of first three years.

Dataset Description:

The dataset used in the project contains CGPA's of 572 students for all the four years including individual grades of all the 42 courses .

Data Preprocessing Steps:

- Import necessary libraries to be used including pandas and numpy .
- Load the dataset by using pandas library using ***pd.read_csv*** built-in function.
- Declare the output of the function . In this dataset the CGPA is the feature which is to be predicted by the model.

```
output=dataset.CGPA
output.head()
```

- Separate the input and output features for training purpose by using ***.drop*** function . In this case CGPA is the output but as the seat number too has no part in output so both the features are to be separated from the input features.

```
courses=dataset.drop(['CGPA','Seat No.'],axis="columns")
courses.info()
```

- The GPs in the given dataset are in grades which needs to be convert in their respective grade points. For this purpose a dictionary of grades with their grade points is to be created and by using for loop over each column in the dataset grade points are assigned to their respective grades.

```
AssignGPs={
    "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,
    "WU" :0
}
```

```
for i in courses:
    courses[i]=courses[i].map(AssignGPs)
courses.head()
```

- After assigning grade points check for the null values in the dataset by using ***isna()*** function. This will print all those features who have any null value in the feature.

```
courses.columns[courses.isna().any()]
```

- If any null values found fill them with 0 using ***.fillna(0)*** function because in data pre-processing we fill empty fields by 0 or average values or used some other technique but according to our dataset we assign it by 0 because we don't give or assign average GPs to student if their data is not entered. Because GPs is different for different student based on their paper.

```
courses[0:]=courses[0:].fillna(0)
```

- Multiply all the GPs of the courses by their respective credithours. Create four arrays containing indexes of the courses according to their credit hours. Iterating a for loop over each index in the array multiply the gradepoints with credithours by using iloc() function. By this function we can easily retrieve any particular value from a row or column using index values.

```
fourcd=[0,5,6,8,9,10,13,18,19,20,21,25,30,31,35,38,39,40]
thirdcd=[2,3,4,7,12,14,15,16,22,23,26,28,32,33,34,37]
secondcd=[1,11,24,27,29,36]
onecd=[17]
for i in fourcd:
  courses.iloc[:, i]=courses.iloc[:, i]*4
for i in thirdcd:
  courses.iloc[:, i]=courses.iloc[:, i]*3
for i in secondcd:
  courses.iloc[:, i]=courses.iloc[:, i]*2

courses.head()
```

Models:

There are three models in the project.
Since we want continuous output which is CGPA so we used all regression algorithms to train our model.

Model no 1:

This model predicts the CGPA based on the courses of first year only. There are two algorithms implemented one is ***linear regression model*** and other is ***random forest regression model***.

A linear regression model describes the relationship between a dependent variable Y and one or more independent variables X. In the given dataset dependent variable is CGPA and independent variables are the courses of first year. So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression.

A random forest is a machine learning technique that's used to solve regression and classification problems. It uses Ensemble learning method which is a technique that combines predictions from

multiple machine learning algorithms to make a more accurate prediction than a single model. 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.

For implementing the algorithms first import both the models from ***sklearn*** library. Then split the dataset into training and testing part by using ***train_test_split()*** function imported from multiple machine learning algorithms to make a more accurate prediction than a single model. 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.

For implementing the algorithms first import both the models from ***sklearn*** library. Then split the dataset into training and testing part by using ***train_test_split()*** function imported from ***sklearn.model_selection*** library. Then predict the models using ***LinearRegression()*** and ***RandomForestRegressor()*** function.

Their training and testing accuracy are:

	Linear Regression	Random Forest Regression
Training Accuracy	0.861	0.978
Testing Accuracy	0.837	0.842

Evaluate the performance of models by ***rmse*** (root mean square error) and **R2_Score** and to prevent the model from overfitting ***cross_val_score*** function is used. Cross Validation prevents from overfitting the model. Since data is of only first year so we get better results but not good or best results.

Model no 2:

This model predicts the CGPA based on the courses of first two years. There are two algorithms implemented one is ***SVM regressor model*** and other is ***KNN regressor model***.

Support Vector Regression is a supervised learning algorithm that is used to predict discrete values. Support Vector Regression uses the same principle as the SVMs. The basic idea behind SVR is to find the best fit line. In SVR, the best fit line is the hyperplane that has the maximum number of points.

The abbreviation KNN stands for "K-Nearest Neighbour". It is a supervised machine learning algorithm. The algorithm can be used to solve both classification and regression problem statements. KNN calculates the distance from all points in the proximity of the unknown data and filters out the ones with the shortest distances to it. As a result, it's often referred to as a distance-based algorithm.

For implementing the algorithms first import both the models from ***sklearn*** library. Then split the dataset into training and testing part by using ***train_test_split()*** function imported from ***sklearn.model_selection*** library. Then predict the models using ***SVR()*** and ***KNeighborsRegressor()*** function.

Their training and testing accuracy are:

	SVM	KNN
Training Accuracy	0.898	0.941
Testing Accuracy	0.953	0.927

Evaluate the performance of models by ***rmse*** (root mean square error) **R2_Score** and to prevent the

model from overfitting ***cross_val_score*** function is used. Cross Validation prevents from overfitting the model.

Model no 3:

This model predicts the CGPA based on the courses of first three years. There are three algorithms implemented one is ***Decision tree model*** and other is ***random forest regression model***.

Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes

For implementing the algorithms first import both the models from ***sklearn*** library. Then split the dataset into training and testing part by using ***train_test_split()*** function imported from ***sklearn.model_selection*** library. Then predict the models using ***DecisionTreeRegressor()*** and ***RandomForestRegressor()*** function.

Their training and testing accuracy are:

	Random Forest	Decision Tree
Training Accuracy	0.992	1.000
Testing Accuracy	0.953	0.823

Evaluate the performance of models by ***rmse*** (root mean square error) **R2_Score** and to prevent the model from overfitting ***cross_val_score*** function is used. Cross Validation prevents from overfitting the model.

Distinguishing features :

There is a user interface in the project which provides the options for selecting between all the three models. After selecting the model it takes input for grades of all the courses . By selecting the submit button it displays the predicted CGPA for final year.

For model 1:

We used Random Forest algorithm model to make prediction because its training and testing accuracy is better than Linear Regression model algorithm model.

For model 2:

We used SVM algorithm model to make prediction because its testing accuracy is better than KNN model algorithm model.

For model 3:

We used Random Forest algorithm model to make prediction because its testing accuracy is better than Decision Tree model algorithm model.

Performance Evaluation:

To evaluate the performance of all the algorithms two evaluation metrics are used. One is ***root mean square error (rmse)*** and other is ***R2 Score.***

The model with the least value of rmse and larger value of r2score is the best model. And both gives values between 0 or 1. Each model rmse and r2score values are listed below:

	MODEL NO 1		MODEL NO 2		MODEL NO 3	
Evaluation metric	Linear regression_1	Random Forest_1	SVM Regressor_2	KNN Regresor_2	Random Forest_3	Decision Tree_3
rmse	0.237	0.2346	0.1454	0.1816	0.1354	0.2642
R2_score	0.837	0.8415	0.9533	0.9271	0.9534	0.8230

according to the above observations it is clear that:

For model 1:

Both algorithm give almost same rmse and r2 values with some minor difference but if we want to choose one Random Forest is best because its rmse value is smaller than Linear Regression's value and its r2 is some decimal points greater than Linear Regression's r2 value.

For model 2:

Here also both algorithm give almost same rmse and r2 values with some minor difference but if we want to choose one SVM is best because its rmse value is smaller (0.04 times) than KNN Regressor's value and its r2 is some decimal points(0.03 times) greater than KNN Regressor's r2 value.

For model 3:

Here Random Forest is the best model than Decision tree because Decision Tree's rmse and r2 value have larger difference(0.1) as compared to Random Forest model which rmse is also very small even smallest than all algorithm's rmse values that we used in whole project and also its r2 have largest value than all algorithm's r2 values that we used in whole project.

Overall Review:

Best predictions are given by model 3 because it has more data (features) than any model which give best training and testing accuracy and in this model best algorithm is Random Forest.

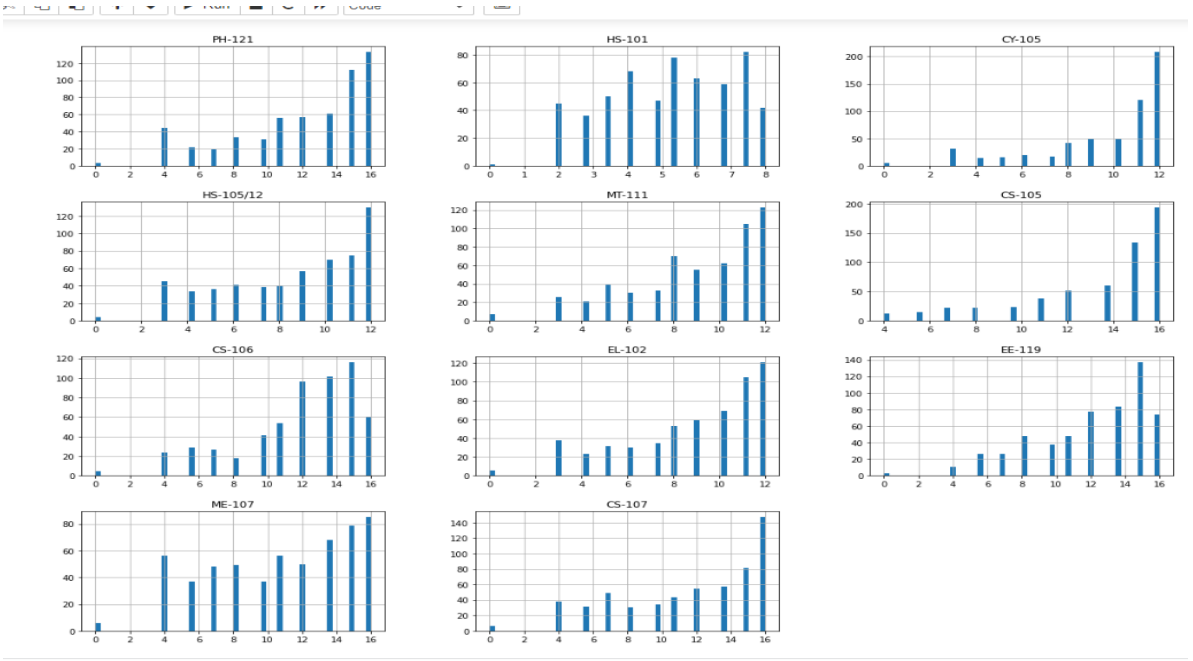
Graphical Comparison:

For graphical comparision we used two graphs:

Histogram:

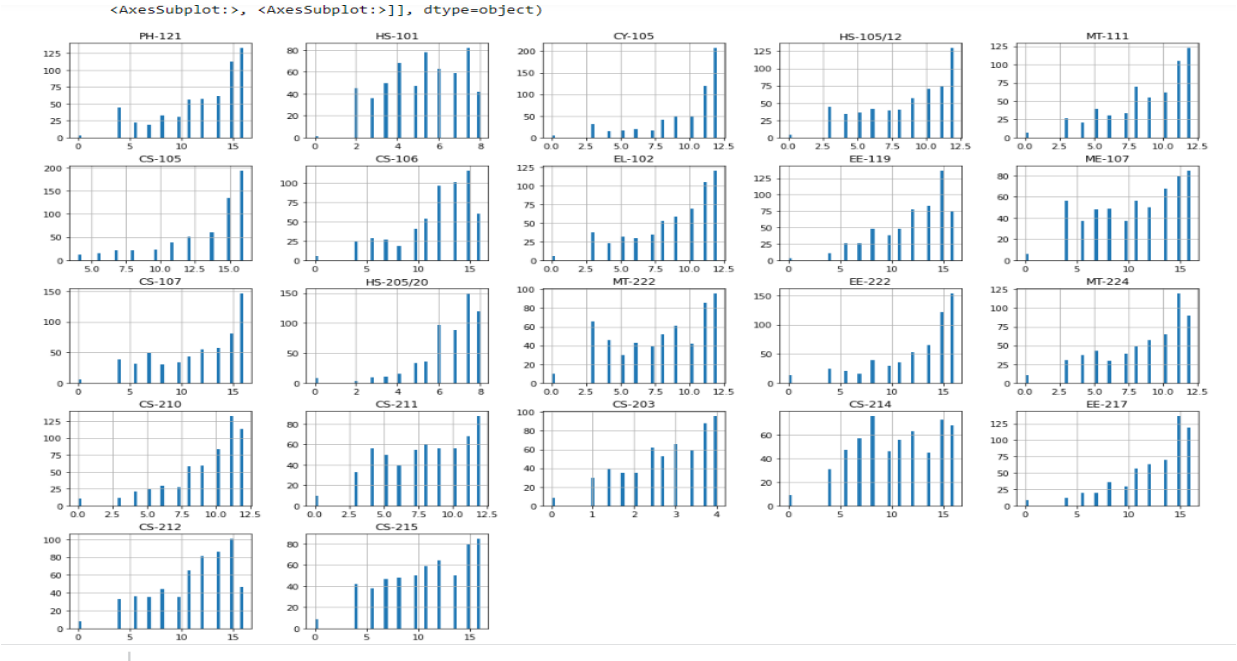
We used histogram to determine how many students score more or how many students score less in each course in terms of their GPs.

For model 1:



It is clearly seen that many students almost 200 students score 4GPA in CS-105 course so we can assume that this course is easy because in it highest number of students score 4GPA as compared to all and in it less student score 1 GPA and almost 60 students score 1GPA in ME-107 course so we can state that this course is difficult because in it highest number of students score 1GPA as compared to all. And they both 4 credit hour course so they effect on CGPA more.

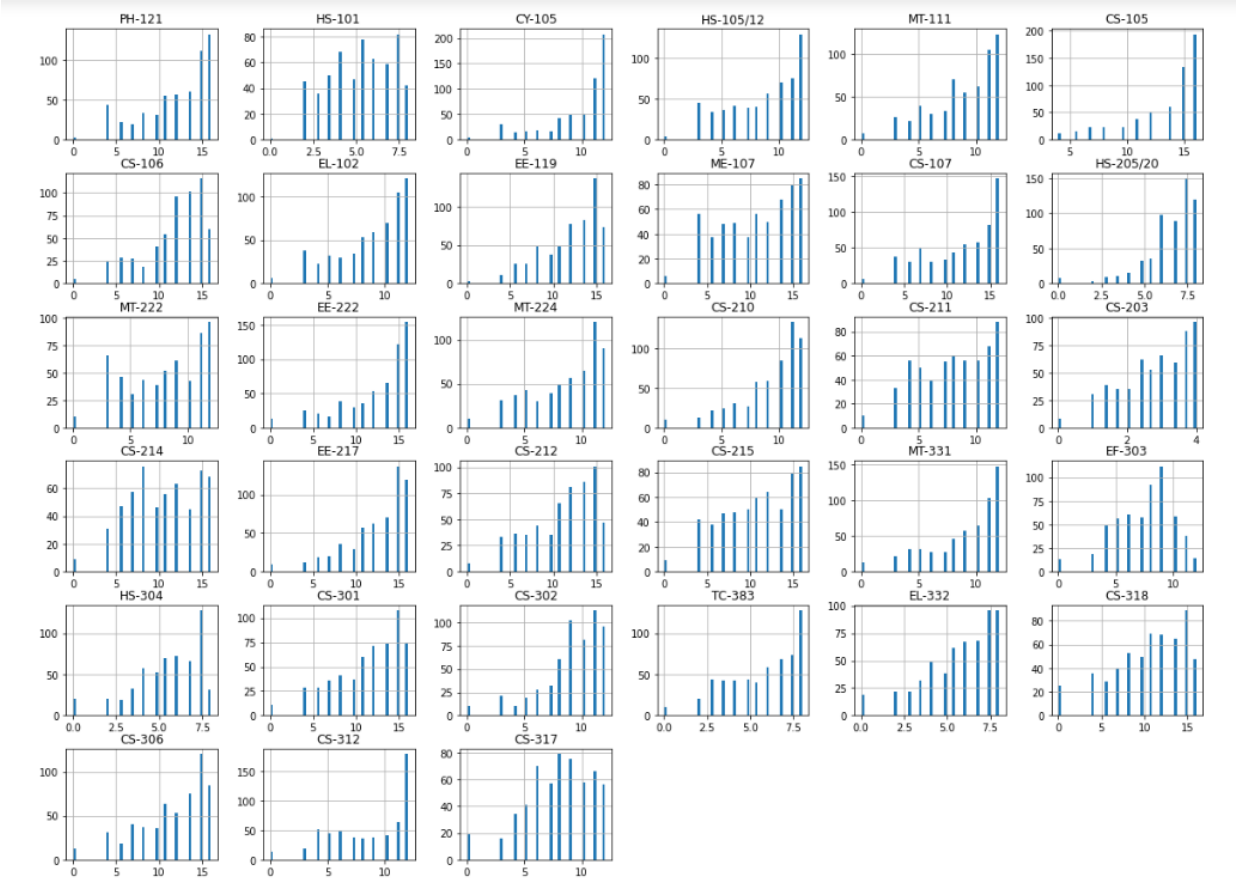
For model 2:



In this also it is clearly seen that many students almost 200 students score 4GPA in CS-105 course so we can assume that this course is easy because in it highest number of students score 4GPA as compared to all and in it less student score 1 GPA and there is one course also CS-107 in which almost 150

students get 4CGPA so this course is also considered as easy course as compared to all and almost 60 students score 1GPA in ME-107 course so we can state that this course is difficult because in it highest number of students score 1GPA as compared to all. And they all are 4 credit hour course so they effect on CGPA more.

For model 3:

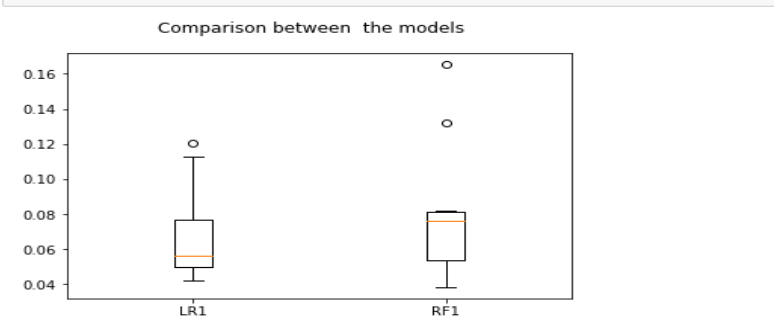


In this CS-105 has more students having 4CGPA and ME-107 has more student having 1CGPA and since in this model we have many courses here so we can predict that in courses with 2 credit hour more student get 4CGPA but it effect less on CGPA because of less credit hour.

Box plot:

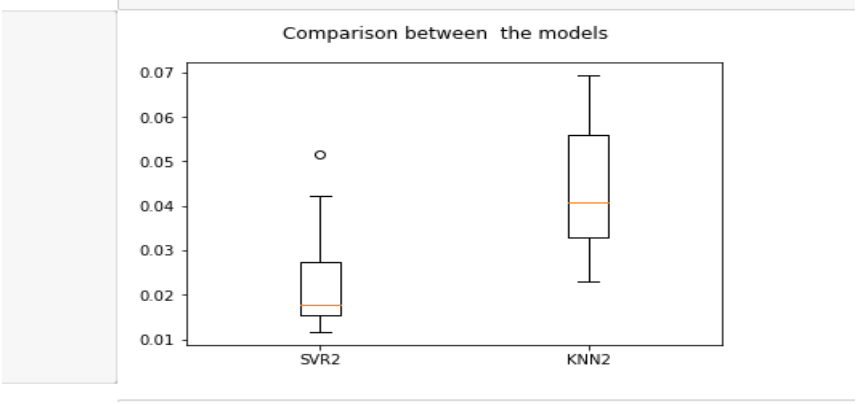
We used this for comparison between models first we do individual comparison of algorithm in each model then we perform overall comparison. We plot this by using cross validation values so we get different values for each folds and observe model in general manner rather than based on only train data or only test data. Since cross validation is calculated in terms of mean squared error(rmse) so the lesser the rmse the better will be the model.

For model 1:



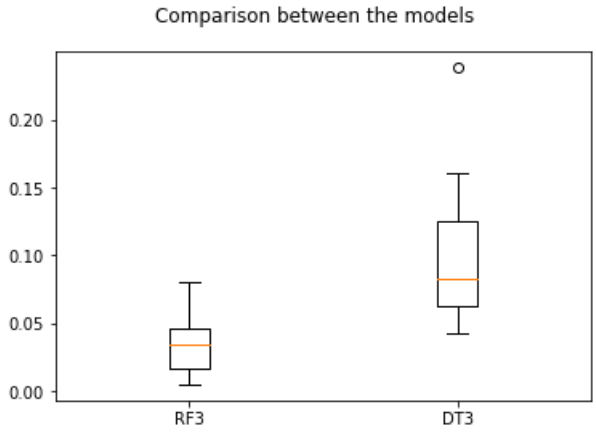
It is clearly visible that Random Forest model is better model because its most cross validation values lies in small range more than 75% or almost 90%..

For model 2:



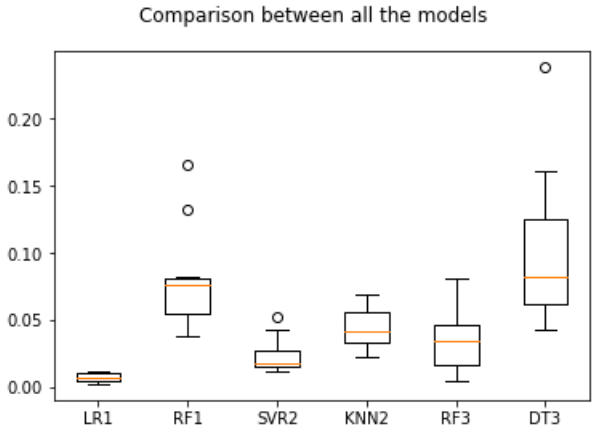
It is clearly visible that SVM model is better model because its most cross validation values lies in small range.

For model 3:



It is clearly visible that Random Forest model is better model because its most cross validation values lies in small range.

Overall Comparison:



In this Random Forest 3 is best model because its more than 75% values lies in lesser range.

Under-fitting and Over-fitting issues:

Under-Fitting:

Since we know that under-fitting is not a big issue as compare to over-fitting but if our data is more under-fitted then our most prediction will be wrong. So in first model we used linear regression model and get some accuracy but its accuracy is not same as we required so to overcome under-fitting we added number of features (courses) which is done in second model so as a test we again train our model with larger number of features as compared to previous one on linear regression algorithm and get more accuracy , this accuracy is good but for getting more we added more courses in third model and gain train our model on linear regression algorithm and get best accuracy.
The accuracy we get by applying linear regression on all models are:

MODELS	ACCURACY(r2_value)
MODEL 1	0.837
MODEL 2	0.9266
MODEL 3	0.9636

So we get best accuracy in model 3 in which only we increased feature and algorithm is same as all.

Over-Fitting:

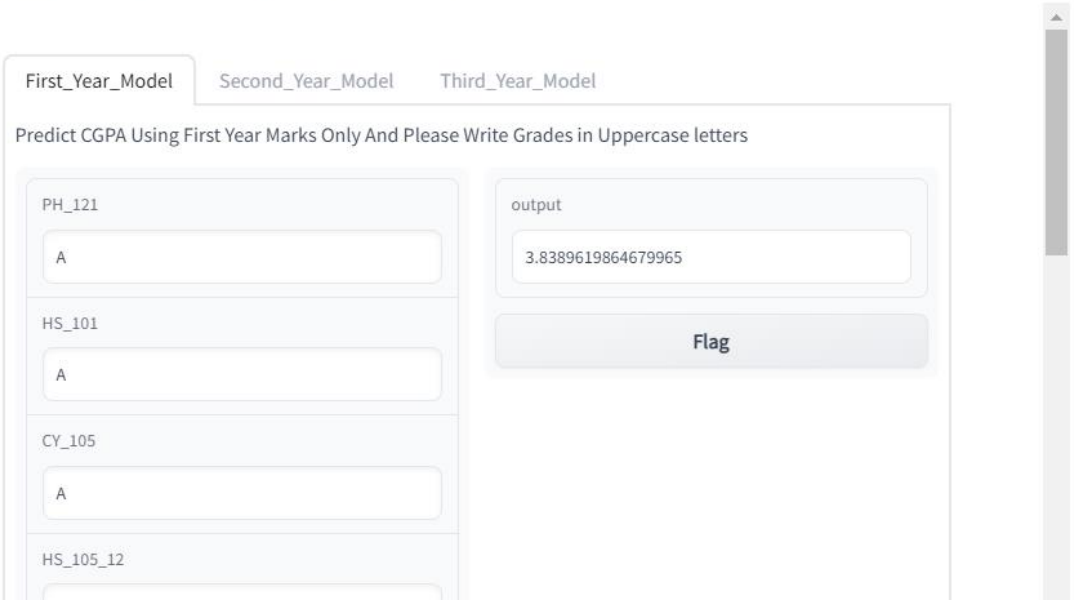
It is a big issue if it happens because it work perfectly on train data set but in test set its predictions are mostly wrong so we cannot rely on predictions from model which have over-fitting issue. So to avoid this issue we used cross-validation in each algorithm for every model in which we train our model in 10 folds so in each fold during training our model have some new data set so now it no more completely rely on fixed train dataset and some old one and in each fold we calculate rmse so in end we get array of rmse and these values are more accurate than individual rmse values of model because here we train our model on whole data set so the difference between its predicted output and expected output is less because by cross-validation we made our model more general.

ALGORITHMS	RMSE	CROSS-VAIDATION RMSE
Linear Regression of model 1	0.237	rmse_of_Linear_Regression_model_1 array([0.19827996, 0.20976113, 0.22540299, 0.19406524, 0.31322986, 0.3811434 , 0.25845866, 0.20162013, 0.27326085, 0.30214621])
Random Forest of model 1	0.2346	array([0.23790559, 0.27414176, 0.28089074, 0.20264099, 0.39251743, 0.40499444, 0.24073409, 0.21616605, 0.278879 , 0.33173])
SVM of model 2	0.1454	array([0.12265985, 0.16298896, 0.17901894, 0.19052359, 0.11631872, 0.15543358, 0.14710213, 0.15927329, 0.14319784, 0.13639336])
KNN of model 2	0.1816	rmse_of_KNeighborsRegressor_model_2 Out[55]: array([0.17541582, 0.18990126, 0.21777961, 0.25789309, 0.12841895, 0.16283559, 0.12196499, 0.22870661, 0.14966023, 0.16895469])
Random Forest of model 3	0.1354	rmse_of_Random_Forest_model_3 array([0.14705665, 0.26379915, 0.12540337, 0.25476757, 0.38222855, 0.14768211, 0.09861294, 0.16705565, 0.17214497, 0.14109086])

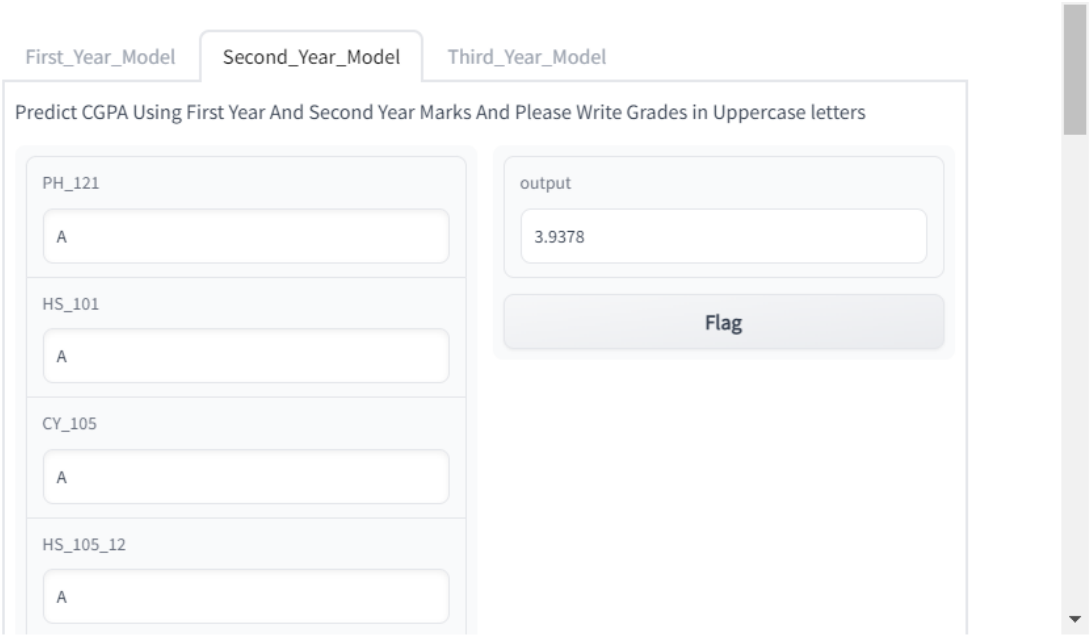
Decision Tree of model 3	0.2642	rmse_of_Decision_Tree_model_3 t[85]: array([0.21646189, 0.37979688, 0.3537717 , 0.27490483, 0.57449775, 0.27135183, 0.22922062, 0.2324408 , 0.30673515, 0.28951998]

OUTPUTS:

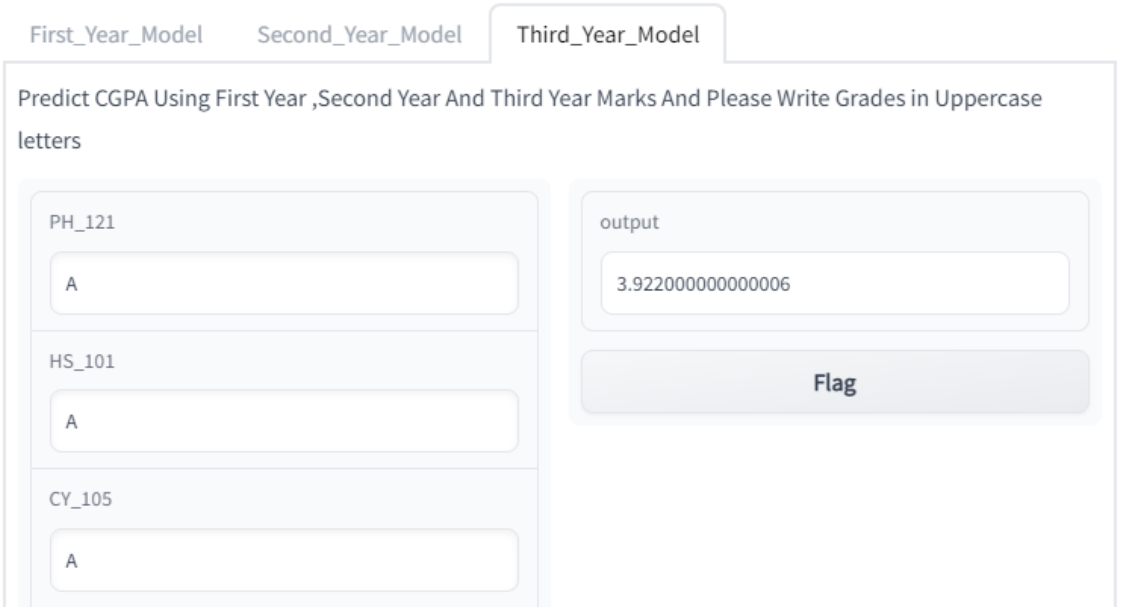
MODEL 1:



MODEL 2:



MODEL 3:


First_Year_Model
Second_Year_Model
Third_Year_Model
Predict CGPA Using First Year ,Second Year And Third Year Marks And Please Write Grades in Uppercase letters
PH_121
A
HS_101
A
CY_105
A
output
3.922000000000006
Flag