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	**MOIZ ALI**	**CS-19087**
S2	**AREEB BIN AZIM**	**CS-19090**
S3	**OMER BIN MUJEEB**	**CS-190095**

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)
= _____

1. Details of the data preprocessing steps applied:

Following steps described the data preprocessing steps:

- The dataset contain GPA's in grade form , So first we convert that grades into actual GPA (decimal form).
- There are some null values in the dataset , So we fill that data via ffill (Forward Fill) or some random GPA.
- Since the whole dataset contain the data till fourth year , So we sample that data to use in Model 1 (till first year) , Model 2 (till Second year) and Model 3 (till Third year) via dropping the unwanted columns.

2. Details of the models and machine algorithm chosen for implementation:

Since the dataset has continous values and the prediction is also a continous value , So we use that two algorithm in our models.

- KNN Regression
- Multiple Linear Regression

KNN regression is a non-parametric method that, in an intuitive manner, approximates the association between independent variables and the continuous outcome by averaging the observations in the same neighbourhood. We set K=5.

Multiple linear regression refers to a statistical technique that uses two or more independent variables to predict the outcome of a dependent variable. The technique enables analysts to determine the variation of the model and the relative contribution of each independent variable in the total variance.

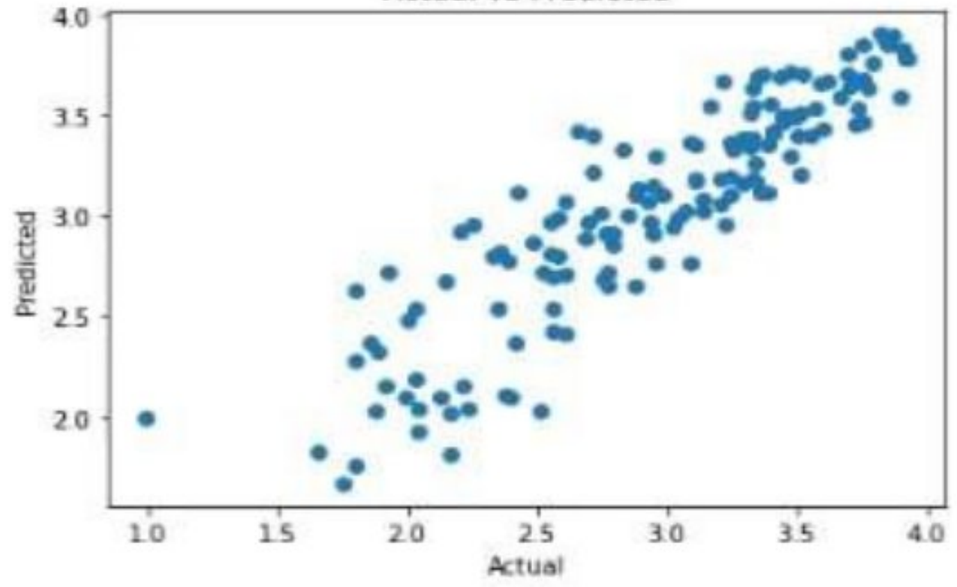
3. Tabular or graphical comparison of all the models:

The score and Graph (Actual vs Predicted) of KNN regression for model1 as follows:

In [32]:
```
accuracy = r2_score(y_test,y_pred)
print("Model Accuracy =", accuracy*100, "in percent")
```

```
Model Accuracy = 77.63536531834183 in percent
```

In [33]:
```
import matplotlib.pyplot as plt
plt.scatter(y_test,y_pred)
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.title('Actual vs Predicted')
plt.show()
```

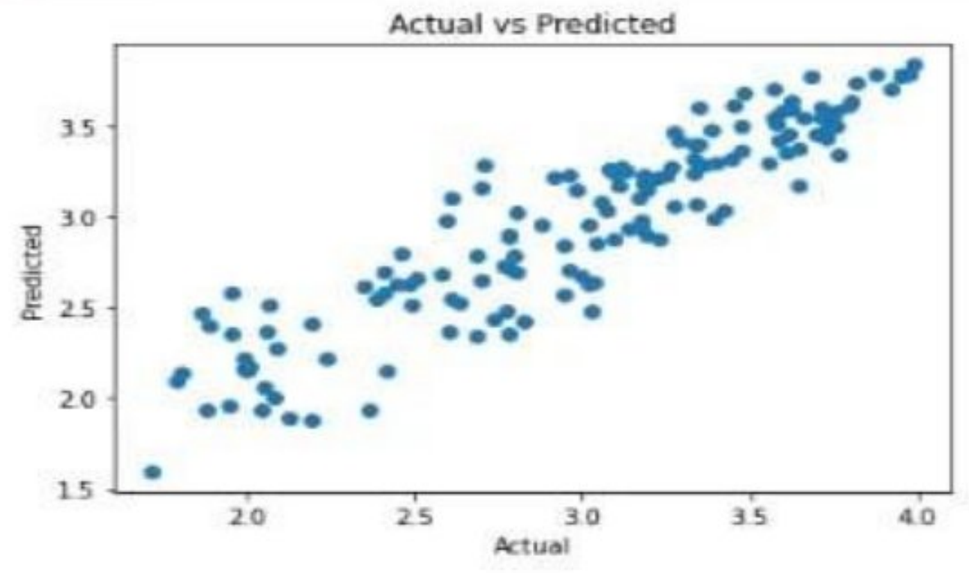


The score and Graph (Actual vs Predicted) of Multiple Linear Regression for model1 as follows:

In [61]:
```
accuracy = r2_score(y_test,y_pred)
print("Model Accuracy =", accuracy*100, "in percent")
```

```
Model Accuracy = 83.58346885831413 in percent
```

In [62]:
```
import matplotlib.pyplot as plt
plt.scatter(y_test,y_pred)
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.title('Actual vs Predicted')
plt.show()
```

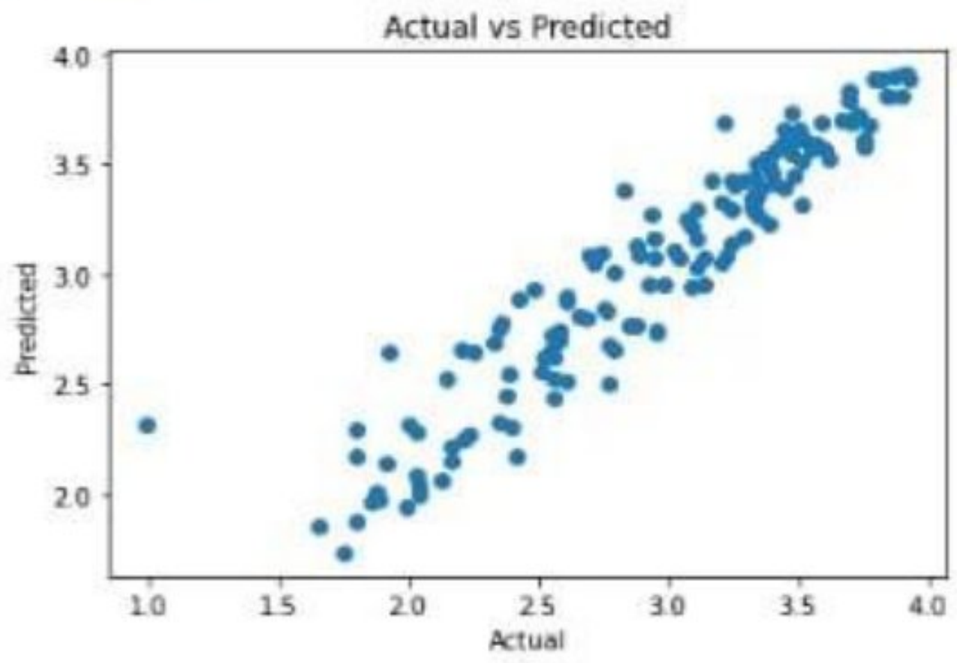


The score and Graph (Actual vs Predicted) of KNN regression for model2 as follows:

In [82]:
```
accuracy = r2_score(y_test,y_pred)
print("Model Accuracy =", accuracy*100, "in percent")
```

```
Model Accuracy = 86.68819209367962 in percent
```

In [83]:
```
import matplotlib.pyplot as plt
plt.scatter(y_test,y_pred)
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.title('Actual vs Predicted')
plt.show()
```



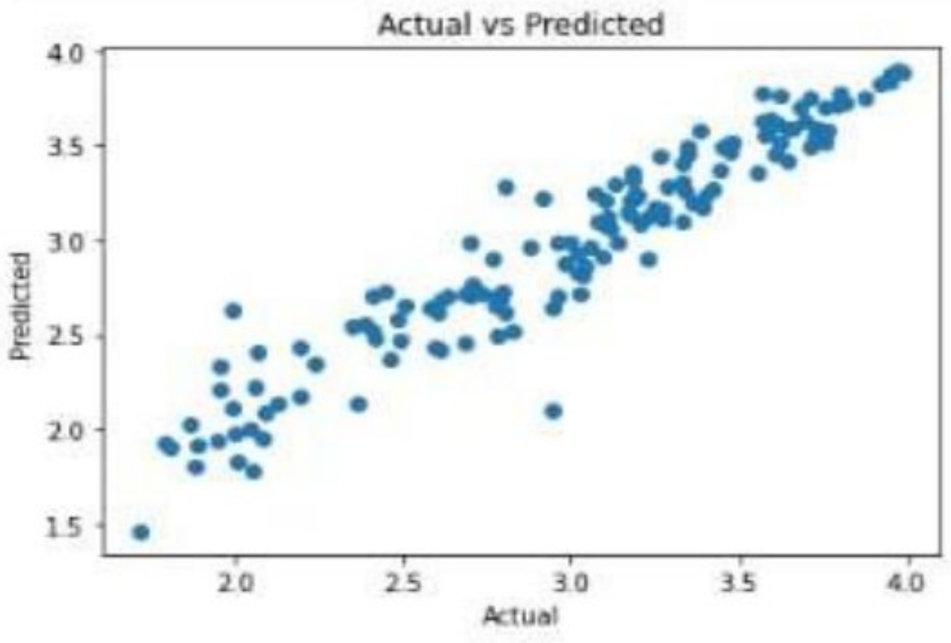
The score and Graph (Actual vs Predicted) of Multiple Linear Regression for model2 as follows:

In [102]:
```
from sklearn.metrics import r2_score
```

In [103]:
```
accuracy = r2_score(y_test,y_pred)
print("Model Accuracy =", accuracy*100, "in percent")
```

```
Model Accuracy = 90.8898535104843 in percent
```

In [104]:
```
import matplotlib.pyplot as plt
plt.scatter(y_test,y_pred)
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.title('Actual vs Predicted')
plt.show()
```



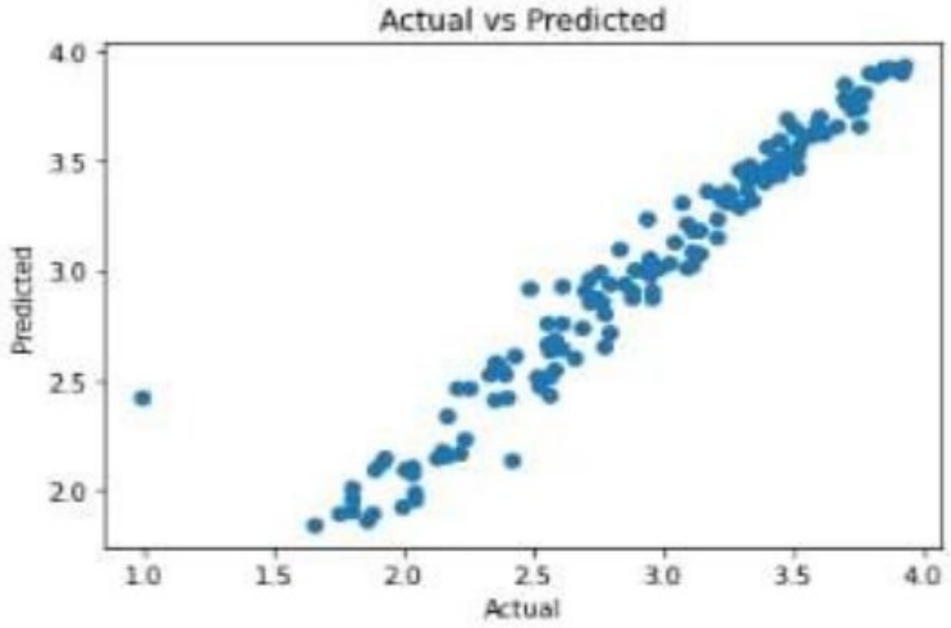
The score and Graph (Actual vs Predicted) of KNN regression for model3 as follows:

In [123]:
```
from sklearn.metrics import r2_score
```

In [124]:
```
accuracy = r2_score(y_test,y_pred)
print("Model Accuracy =", accuracy*100, "in percent")
```

```
Model Accuracy = 92.09891623295616 in percent
```

In [125]:
```
import matplotlib.pyplot as plt
plt.scatter(y_test,y_pred)
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.title('Actual vs Predicted')
plt.show()
```



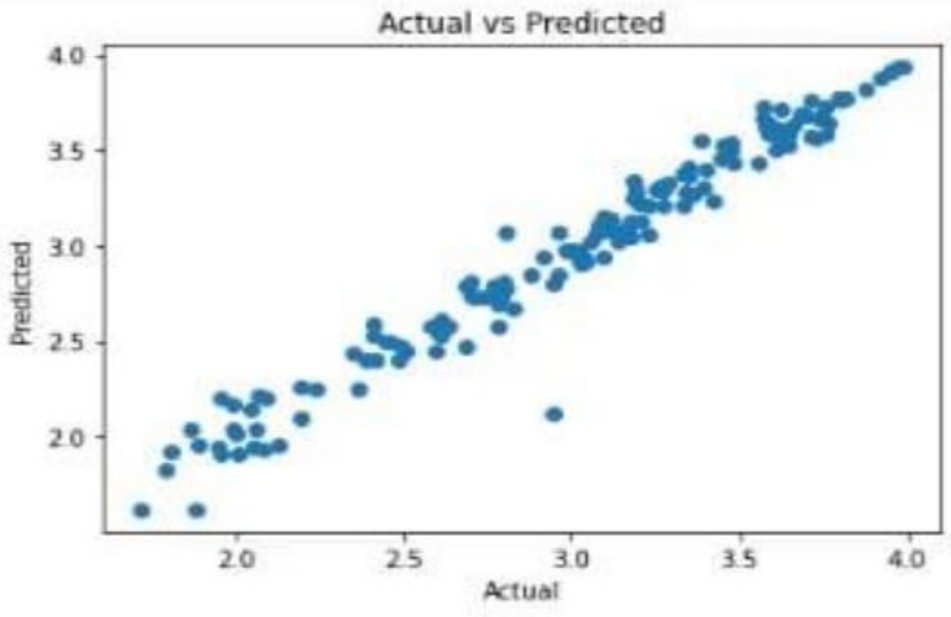
The score and Graph (Actual vs Predicted) of Multiple Linear Regression for model3 as follows:

In [144]:
```
from sklearn.metrics import r2_score
```

In [146]:
```
accuracy = r2_score(y_test,y_pred)
print("Model Accuracy =", accuracy*100, "in percent")
```

```
Model Accuracy = 96.07720030539976 in percent
```

In [147]:
```
import matplotlib.pyplot as plt
plt.scatter(y_test,y_pred)
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.title('Actual vs Predicted')
plt.show()
```



4. Comments on the performance of the implemented machine learning system:

The accuracy of KNN Regression is better than the Multiple Linear Regression. Its because of the reason that KNN stores all available cases and classifies new cases based on a similarity measure. The KNN algorithm assumes that similar things exist in close proximity. While the Multiple linear regression model allows an analyst to predict an outcome based on information provided on multiple explanatory variables. Still, the model is not always perfectly accurate as each data point can differ slightly from the outcome predicted by the model.

5. Suggestions:

- We have used some random GPA's in order to fill the null data, if we use any average value or median of the values. So the performance of the model could be enhanced.
- Adding more data is a good suggestion in order to improve the algorithm efficiency.
- Hitting at the right machine learning algorithm is the ideal approach to achieve higher accuracy. So we should use Multiple algorithms to best fit the model.