**PROJECT TITLE**

**A MINI PROJECT REPORT**

**18CSC305J - ARTIFICIAL INTELLIGENCE**

### Submitted by

**Ruthwik Reddy[RA2011003466]**

*Under the guidance of*

**DR.M.ARUNA**

### Assistant Professor, Department of Computer Science and Engineering

### in partial fulfillment for the award of the degree of

**BACHELOR OF TECHNOLOGY**

in

**COMPUTER SCIENCE & ENGINEERING**

of

**FACULTY OF ENGINEERING AND TECHNOLOGY**



#### S.R.M. Nagar, Kattankulathur, Chengalpattu District

**MAY 2023**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

(Under Section 3 of UGC Act, 1956)

**BONAFIDE CERTIFICATE**

Certified that Mini project report titled **“STUDENT-GRADE ANALYSIS”** is the bona fide work of **RUTHWIK REDDY (RA2011003010466)** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**SIGNATURE SIGNATURE**

|  |  |
| --- | --- |
| GUIDE NAME  **GUIDE**  Assistant Professor  Department of Computing Technologies | Dr. M. Pushpalatha  **HEAD OF THE DEPARTMENT**  Professor & Head  Department of Computing Technologies |

**ABSTRACT**

In the world of open education systems, students have flexibility to learn anything with ease as the learning content is easily available. But this facility can make student complacent. Therefore, it becomes difficult to predict the student’s performance in advance. In this project, an attempt is made to help the student to know his performance in advance. This is done by using univariate linear regression model. This would help students to improve their performance based on predicted grades and would enable teachers to identify those individuals who need assistance. The Main Objective of “Student Grade Analysis and Prediction” is to implement a simple algorithmic model that predicts the score of an individual student at he /she end of the year

**TABLE OF CONTENTS**

**[ABSTRACT](#_bookmark1)** **iii**

**TABLE OF CONTENTS iv**

**[LIST OF FIGURES](#_bookmark2) v**

**[ABBREVIATIONS](#_bookmark3) vi**

**[1 INTRODUCTION](#_bookmark4) 7**

**[2 LITERATURE SURVEY](#_bookmark20) 8**

**3 [SYSTEM A](#_bookmark30)RCHITECTURE AND DESIGN 9**

3.1 Architecture diagram of student grade analysis 9

3.2 Description of Module and components10

**4 METHODOLOGY 14**

4.1 Methodological Steps 14

**5 CODING AND TESTING 15**

**6 SREENSHOTS AND RESULTS**

6.1 Student Grade analysis chart 19

6.2 student grade analysis flowchart 19

6.3 student grade analysis code 20

6.4 Finding mean absolute error 20

6.5 Finding Root Mean square error 20

**7 CONCLUSION AND FUTURE ENHANCEMENT 23**

7.1 Conclusion

7.2 Future Enhancement

**REFERENCES 24**

**LIST OF FIGURES**

2  **Literature survey**  7

##### [3.1.1](#_bookmark12) Architecture block 9

[3.2.1](#_bookmark16) **Sytem Design** 10

[6.1.1](#_bookmark43) **screenshot of plots of student grade prediction** 19

[6.3.1](#_bookmark52) **Screenshot of flowchart student grade prediction** 20

[6.4.1](#_bookmark54) **screenshot of codes** 20

6.6.1 **Screenshot of results** 21

7.1.1 **To Find Mean Absolute Error** 22

7.2.1 **To Find Root Mean Squared Error 22**

v

**ABBREVIATIONS**

**EDM** Education data mining

**MAE** Mean Absolute Error

**RMSE** Root Mean Square Error

**SGP** Student grade prediction

**OER**  Open Education Resources

**vi**

**CHAPTER 1**

**INTRODUCTION**

In the present educational systems, student performance is getting worsen everyday gradually . Predicting student performance in advance can help students as well as their teachers to keep track of progress of the student. Many educational institutes have adopted continuous evaluation system today. Such systems are favourable to the students in improving their performance . The purpose of the continuous evaluation system is to help the regular students in their academics. In continuous evaluation system, unit tests or class tests are conducted at regular period. To have consistent performance in the final grade it is required to appear in all the unit tests or class test. The core function of Student Grade Prediction is to help the student to know his/her performance in advance by using uni variance Linear Regression Model. Such techniques would help the students to improve their performance based on the predicted grade and would enable teachers to identify those individuals who might need assistance.

**CHAPTER 2**

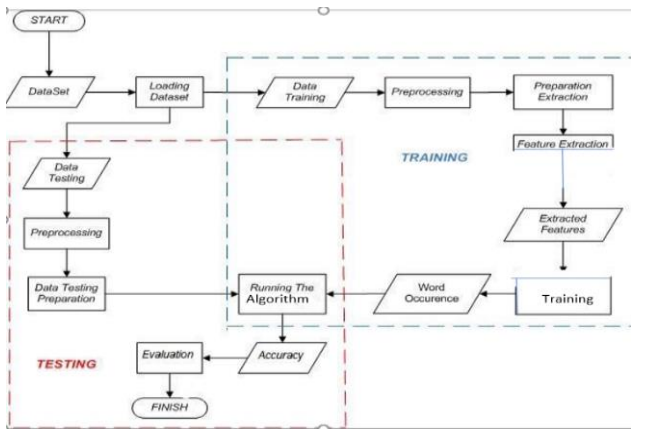
**LITERATURE SURVEY**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SR.NO** | **TITLE** | **Author Name and Year of Publication** | **Methodology** | **Inference** | **Drawbacks** |
| **1.** | **Literature survey**  **on student's**  **performance prediction in**  **education using**  **data mining**  **techniques.** | **Kumar M,2017** | **students performance prediction** | **Important factors**  **of students used**  **For predicting**  **Students performance**  **Accuracy of Different**  **prediction models** |  |
| **2.** | **survey on**  **Various aspects of education data**  **mining in**  **Predicting student performance** | **Shingari I,2018** | **predicting students performance** | **Education Data Mining (EDM) is an interdisciplinary**  **straightforward field to investigate that handles the**  **Advancement of**  **strategies** |  |
| **3.** | **A machine**  **learning algorithm framework for**  **predicting students performance** | **Qazdar A,2018** | **predicting students perfomance by machine learning** | **we present a**  **framework for**  **predicting student**  **performance based on Machine Learning**  **algorithm** |  |
| **4.** | **systematic review on educational**  **data mining.** | **Dutt A,2018** | **educational data mining** | **This implies that a preprocessing algorithm has to be enforced first and only then some specific data mining methods can be applied to the problems** |  |

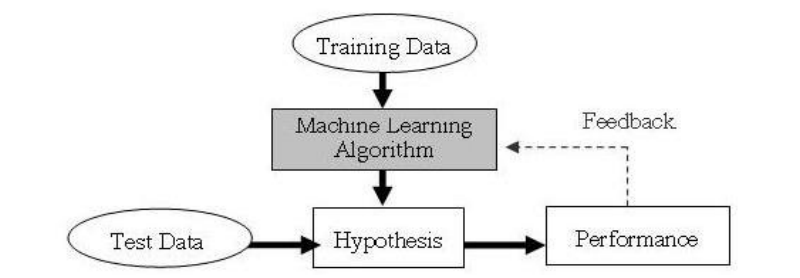
**CHAPTER 3**

**SYSTEM ARCHITECTURE AND DESIGN**

**DESIGN**

****

**SYSTEM ARCHITECTURE**

****

**CHAPTER 4**

**METHODOLOGY**

Since universities are prestigious places of higher education, students’ retention in these universities is a matter of high concern. It has been found that most of the students’ drop-out from the universities during their first year is due to lack of proper support in undergraduate courses. Due to this reason, the first year of the undergraduate student is referred as a “make or break” year. Without getting any support on the course domain and its complexity, it may demotivate a student and can be the cause to withdraw the course.

There is a great need to develop an appropriate solution to assist students retention at higher education institutions. Early grade prediction is one of the solutions that have a tendency to monitor students’ progress in the degree courses at the University and will lead to improving the students’ learning process based on predicted grades.

Using machine learning with Educational Data Mining can improve the learning process of students. Different models can be developed to predict students’ grades in the enrolled courses, which provide valuable information to facilitate students’ retention in those courses. This information can be used to early identify students at-risk based on which a system can 1 suggest the instructors to provide special attention to those students. This information can also help in predicting the students’ grades in different courses to monitor their performance in a better way that can enhance the students’ retention rate of the universities.

Using various packages such as cufflink, seaborn & matplotlib to represent the data along with different attributes graphically or pictorially to analyse the dataset for predicting the Final Grade(G3).

**CHAPTER 5**

**CODING AND TESTING**

**import** pandas **as** pd

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**import** seaborn **as** sns

**%matplotlib** inline

stud**=** pd**.**read\_csv('student-mat.csv') *# Read the dataset*

In [3]:

print('Total number of students:',len(stud))

stud['G3']**.**describe()

stud**.**info() *# Information on dataset*

stud**.**columns *# Dataset Columns*

stud**.**describe() *# Dataset description*

stud**.**head() *# First 5 values of dataset*

stud**.**tail() *# Last 5 values of dataset*

stud**.**isnull()**.**any() *# To check any null values present in dataset*

**import** cufflinks **as** cf

cf**.**go\_offline()

stud**.**iplot() *# Plot for the all attributes*

In [13]:

stud**.**iplot(kind**=**'scatter',x**=**'age',y**=**'G3',mode**=**'markers',size**=**8) *# Plot for age vs G3*

In [14]:

stud**.**iplot(kind**=**'box')

In [15]:

stud['G3']**.**iplot(kind**=**'hist',bins**=**100,color**=**'blue')

sns**.**heatmap(stud**.**isnull(),cmap**=**"rainbow",yticklabels**=False**) *# To check any null values present in dataset pictorially*

sns**.**heatmap(stud**.**isnull(),cmap**=**"viridis",yticklabels**=False**) *# Map color - viridis*

f\_stud **=** len(stud[stud['sex'] **==** 'F']) *# Number of female students*

print('Number of female students:',f\_stud)

m\_stud **=** len(stud[stud['sex'] **==** 'M']) *# Number of male students*

print('Number of male students:',m\_stud)

sns**.**set\_style('whitegrid') *# male & female student representaion on countplot*

sns**.**countplot(x**=**'sex',data**=**stud,palette**=**'plasma')

b **=** sns**.**kdeplot(stud['age']) *# Kernel Density Estimations*

b**.**axes**.**set\_title('Ages of students')

b**.**set\_xlabel('Age')

b**.**set\_ylabel('Count')

plt**.**show()

b **=** sns**.**countplot(x**=**'age',hue**=**'sex', data**=**stud, palette**=**'inferno')

b**.**axes**.**set\_title('Number of Male & Female students in different age groups')

b**.**set\_xlabel("Age")

b**.**set\_ylabel("Count")

plt**.**show()

u\_stud **=** len(stud[stud['address'] **==** 'U']) *# Number of urban areas students*

print('Number of Urban students:',u\_stud)

r\_stud **=** len(stud[stud['address'] **==** 'R']) *# Number of rural areas students*

print('Number of Rural students:',r\_stud)

sns**.**set\_style('whitegrid')

sns**.**countplot(x**=**'address',data**=**stud,palette**=**'magma') *# urban & rural representaion on countplot*

*# Grade distribution by address*

sns**.**kdeplot(stud**.**loc[stud['address'] **==** 'U', 'G3'], label**=**'Urban', shade **=** **True**)

sns**.**kdeplot(stud**.**loc[stud['address'] **==** 'R', 'G3'], label**=**'Rural', shade **=** **True**)

plt**.**title('Do urban students score higher than rural students?')

plt**.**xlabel('Grade');

plt**.**ylabel('Density')

plt**.**show()

**from** sklearn.preprocessing **import** LabelEncoder

le**=**LabelEncoder()

stud**.**iloc[:,0]**=**le**.**fit\_transform(stud**.**iloc[:,0])

stud**.**iloc[:,1]**=**le**.**fit\_transform(stud**.**iloc[:,1])

stud**.**iloc[:,3]**=**le**.**fit\_transform(stud**.**iloc[:,3])

stud**.**iloc[:,4]**=**le**.**fit\_transform(stud**.**iloc[:,4])

stud**.**iloc[:,5]**=**le**.**fit\_transform(stud**.**iloc[:,5])

stud**.**iloc[:,8]**=**le**.**fit\_transform(stud**.**iloc[:,8])

stud**.**iloc[:,9]**=**le**.**fit\_transform(stud**.**iloc[:,9])

stud**.**iloc[:,10]**=**le**.**fit\_transform(stud**.**iloc[:,10])

stud**.**iloc[:,11]**=**le**.**fit\_transform(stud**.**iloc[:,11])

stud**.**iloc[:,15]**=**le**.**fit\_transform(stud**.**iloc[:,15])

stud**.**iloc[:,16]**=**le**.**fit\_transform(stud**.**iloc[:,16])

stud**.**iloc[:,17]**=**le**.**fit\_transform(stud**.**iloc[:,17])

stud**.**iloc[:,18]**=**le**.**fit\_transform(stud**.**iloc[:,18])

stud**.**iloc[:,19]**=**le**.**fit\_transform(stud**.**iloc[:,19])

stud**.**iloc[:,20]**=**le**.**fit\_transform(stud**.**iloc[:,20])

stud**.**iloc[:,21]**=**le**.**fit\_transform(stud**.**iloc[:,21])

stud**.**iloc[:,22]**=**le**.**fit\_transform(stud**.**iloc[:,22])

*# Standard ML Models for comparison*

**from** sklearn.linear\_model **import** LinearRegression

**from** sklearn.linear\_model **import** ElasticNet

**from** sklearn.ensemble **import** RandomForestRegressor

**from** sklearn.ensemble **import** ExtraTreesRegressor

**from** sklearn.ensemble **import** GradientBoostingRegressor

**from** sklearn.svm **import** SVR

*# Splitting data into training/testing*

**from** sklearn.model\_selection **import** train\_test\_split

**from** sklearn.preprocessing **import** MinMaxScaler

*# Metrics*

**from** sklearn.metrics **import** mean\_squared\_error, mean\_absolute\_error, median\_absolute\_error

*# Distributions*

**import** scipy

*# splitting the data into training and testing data (75% and 25%)*

*# we mention the random state to achieve the same split everytime we run the code*

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(stud, stud['G3'], test\_size **=** 0.25, random\_state**=**42)

X\_train**.**head()

*# Calculate mae and rmse*

**def** evaluate\_predictions(predictions, true):

mae **=** np**.**mean(abs(predictions **-** true))

rmse **=** np**.**sqrt(np**.**mean((predictions **-** true) **\*\*** 2))

**return** mae, rmse

In [48]:

*# find the median*

median\_pred **=** X\_train['G3']**.**median()

*# create a list with all values as median*

median\_preds **=** [median\_pred **for** \_ **in** range(len(X\_test))]

*# store the true G3 values for passing into the function*

true **=** X\_test['G3']

In [49]:

*# Display the naive baseline metrics*

mb\_mae, mb\_rmse **=** evaluate\_predictions(median\_preds, true)

print('Median Baseline MAE: {:.4f}'**.**format(mb\_mae))

print('Median Baseline RMSE: {:.4f}'**.**format(mb\_rmse))

*# Evaluate several ml models by training on training set and testing on testing set*

**def** evaluate(X\_train, X\_test, y\_train, y\_test):

*# Names of models*

model\_name\_list **=** ['Linear Regression', 'ElasticNet Regression',

'Random Forest', 'Extra Trees', 'SVM',

'Gradient Boosted', 'Baseline']

X\_train **=** X\_train**.**drop('G3', axis**=**'columns')

X\_test **=** X\_test**.**drop('G3', axis**=**'columns')

*# Instantiate the models*

model1 **=** LinearRegression()

model2 **=** ElasticNet(alpha**=**1.0, l1\_ratio**=**0.5)

model3 **=** RandomForestRegressor(n\_estimators**=**100)

model4 **=** ExtraTreesRegressor(n\_estimators**=**100)

model5 **=** SVR(kernel**=**'rbf', degree**=**3, C**=**1.0, gamma**=**'auto')

model6 **=** GradientBoostingRegressor(n\_estimators**=**50)

*# Dataframe for results*

results **=** pd**.**DataFrame(columns**=**['mae', 'rmse'], index **=** model\_name\_list)

*# Train and predict with each model*

**for** i, model **in** enumerate([model1, model2, model3, model4, model5, model6]):

model**.**fit(X\_train, y\_train)

predictions **=** model**.**predict(X\_test)

*# Metrics*

mae **=** np**.**mean(abs(predictions **-** y\_test))

rmse **=** np**.**sqrt(np**.**mean((predictions **-** y\_test) **\*\*** 2))

*# Insert results into the dataframe*

model\_name **=** model\_name\_list[i]

results**.**loc[model\_name, :] **=** [mae, rmse]

*# Median Value Baseline Metrics*

baseline **=** np**.**median(y\_train)

baseline\_mae **=** np**.**mean(abs(baseline **-** y\_test))

baseline\_rmse **=** np**.**sqrt(np**.**mean((baseline **-** y\_test) **\*\*** 2))

results**.**loc['Baseline', :] **=** [baseline\_mae, baseline\_rmse]

**return** results

In [51]:

results **=** evaluate(X\_train, X\_test, y\_train, y\_test)

results

plt**.**figure(figsize**=**(12, 7))

*# Root mean squared error*

ax **=** plt**.**subplot(1, 2, 1)

results**.**sort\_values('mae', ascending **=** **True**)**.**plot**.**bar(y **=** 'mae', color **=** 'violet', ax **=** ax)

plt**.**title('Model Mean Absolute Error')

plt**.**ylabel('MAE')

*# Median absolute percentage error*

ax **=** plt**.**subplot(1, 2, 2)

results**.**sort\_values('rmse', ascending **=** **True**)**.**plot**.**bar(y **=** 'rmse', color **=** 'pink', ax **=** ax)

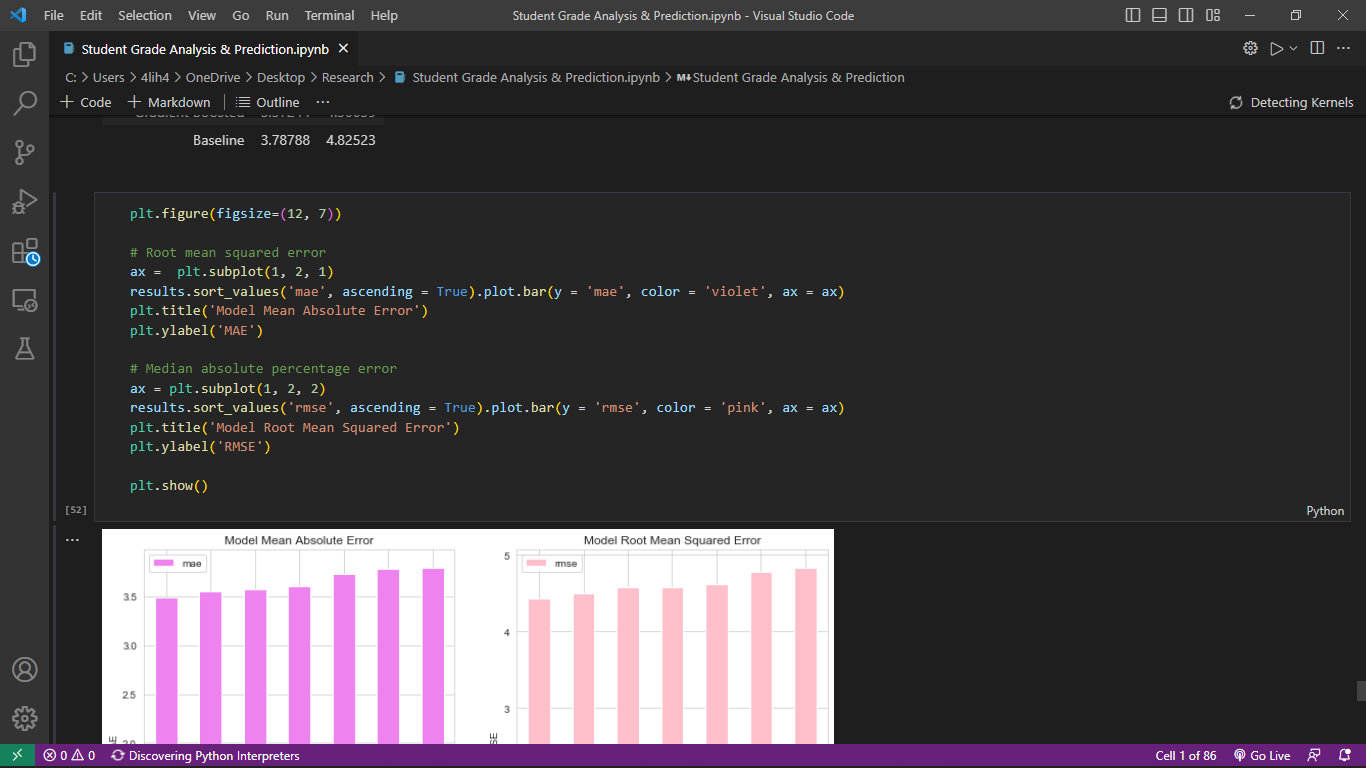
plt**.**title('Model Root Mean Squared Error')

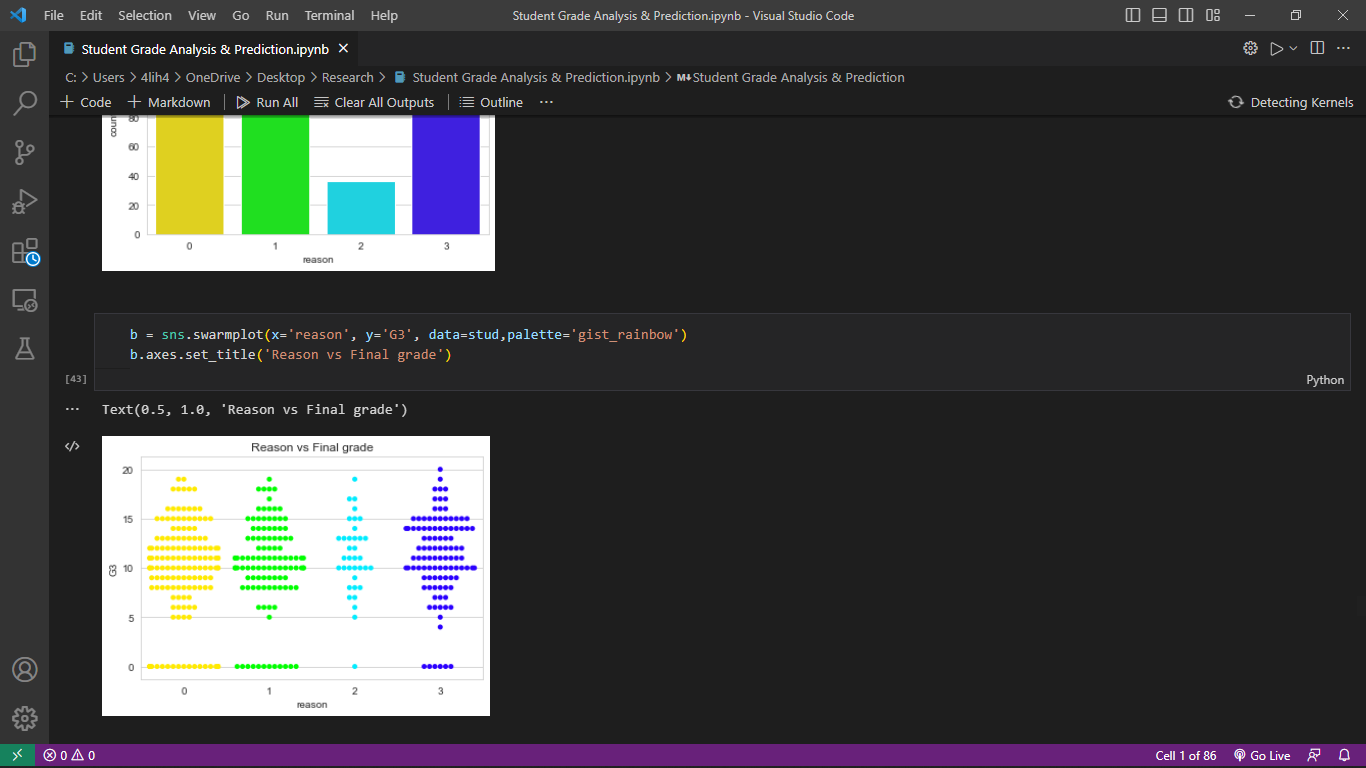
plt**.**ylabel('RMSE')

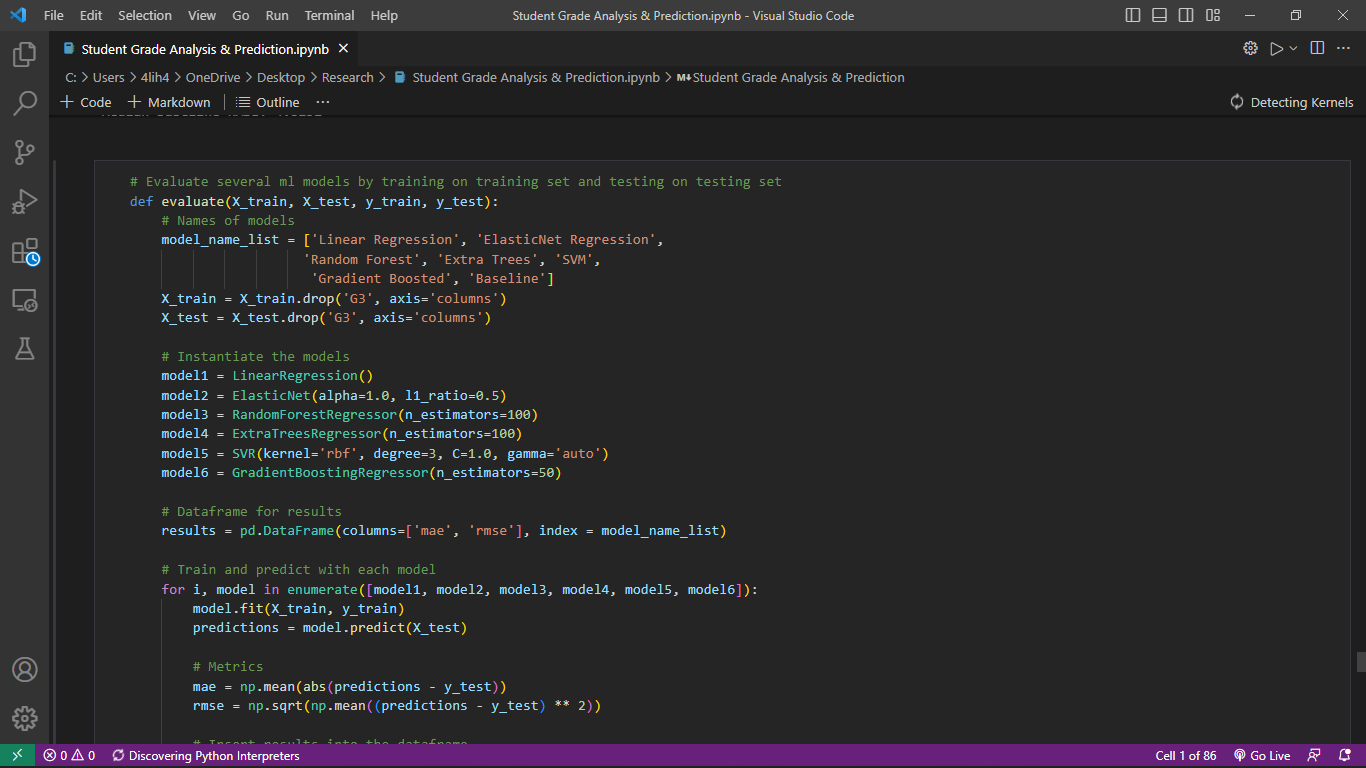
plt**.**show()

**CHAPTER 6**

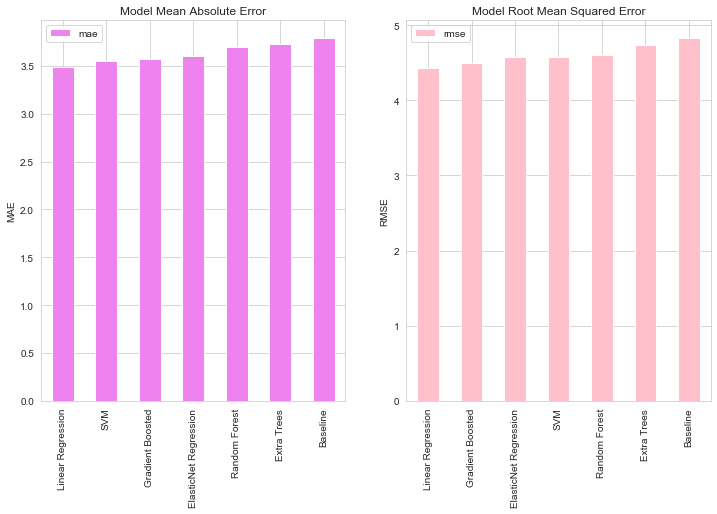
**SCREENSHOTS AND RESULTS**

****

****

**CHAPTER 7**

**CONCLUSION AND FUTURE ENHANCEMENTS**

As we see both MAE & Model RMSE that the Linear Regression is performing the best in both cases.

Data cleaning and analysis can be better done and other machine learning algorithms can be applied on the model to improve the accuracy.Increased dataset will give out more accurate predictions. To improve the results, a dataset with sufficient features and increase in quantity must be obtained. Further research must be conducted in enhancing the existing machine learning techniques to work in real time and develop an efficient model. Also, the models developed must be tested on data with different volumes to test its scalability and performance. In future work, the result of regression on balanced dataset can be studied by changing the data distribution. This can be done by selecting a sample of dataset or removing certain records to balance the type of data.

**REF****ERENCES**

1. Alshammari, R., & Khan, S. U. (2018). Machine Learning Techniques for Educational Data Mining: A Survey. IEEE Access, 6, 53409-53427.

2. Bawaneh, S. S. (2019). A Comparative Study on Classification Models for Predicting Student Performance. Journal of Educational Computing Research, 57(5), 1139-1159.

3. Chakraborty, T., Chakraborty, S., & Das, R. (2021). Predictive analytics of student academic performance using machine learning: A review. Journal of King Saud University-Computer and Information Sciences, 33(1), 77-89.