CISC5790 Data Mining Course

Graduate Level Course Project Presentation
Dec 4, 2020 I Fordham University I New York

Undergraduate GPA Prediction using Python-based Data Mining and Machine Learning Approaches: A Comparative Assessment

Vaishali Sharma, Graduate Data Science Student, Fordham University

Project Advisor

Prof. Gary Weiss
Department of Computer and Information Science
Fordham University, New York

Motivation

- Identify students at risk and provide adequate advising and tailored help towards reduced failure rates.
- Detecting high and low achiever students and help them enhance their career paths.
- Predictors and Early Warning Systems can be developed
- A predictor “given a specific set of input data, aims to anticipate the outcomes of a course or a degree”
- EWS reports finding to the teachers or students at early stage so that measures can be taken to avoid the negative outcomes.

Broad Objective:

Undergraduate GPA Prediction using Python-based Data Mining and Machine Learning Approaches: A Comparative Assessment of all the models

Project Outline and Methods

- Data Imputation
- Undergraduate final GPA prediction using all course term GPA
- Develop a relationship between first two Terms and Final GPA using Python-based Data Mining tools and Machine learning algorithms.
- Predict future GPA using developed algorithms
- Model Evaluation and Validation
- Accuracy Evaluation

Python-Based Models

- Linear Regression
- SVR
- K-Nearest Neighbor
- NB

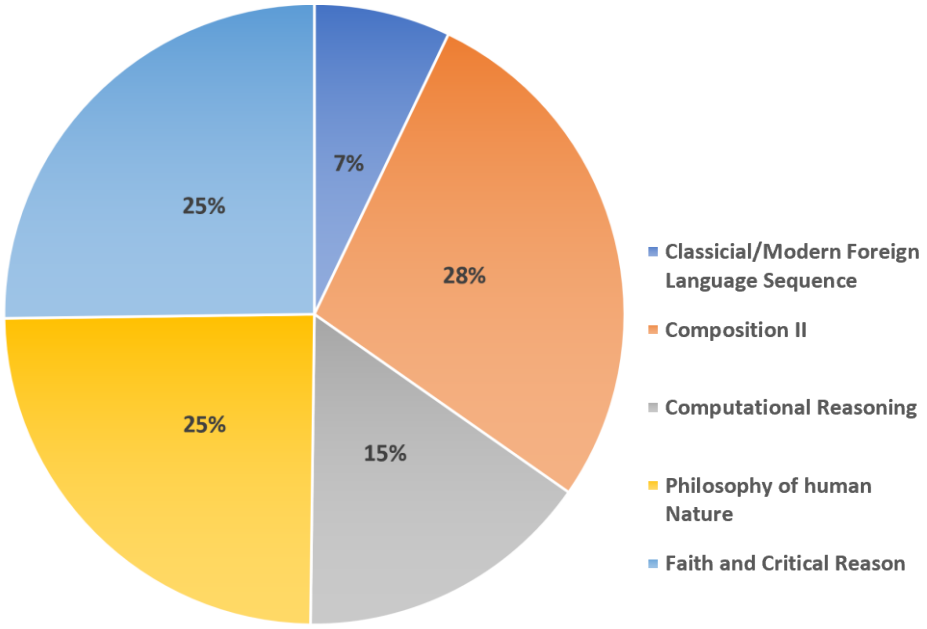
Python Libraries

certifi==2020.11.8; contextlib2==0.6.0.post1; Cython==0.29.20;
joblib==0.17.0; lxml==4.5.1; numpy==1.19.4; pandas==1.1.4
Pillow==7.1.2; python-dateutil==2.8.1; pytz==2020.4; scikit-learn==0.23.2

Data Description

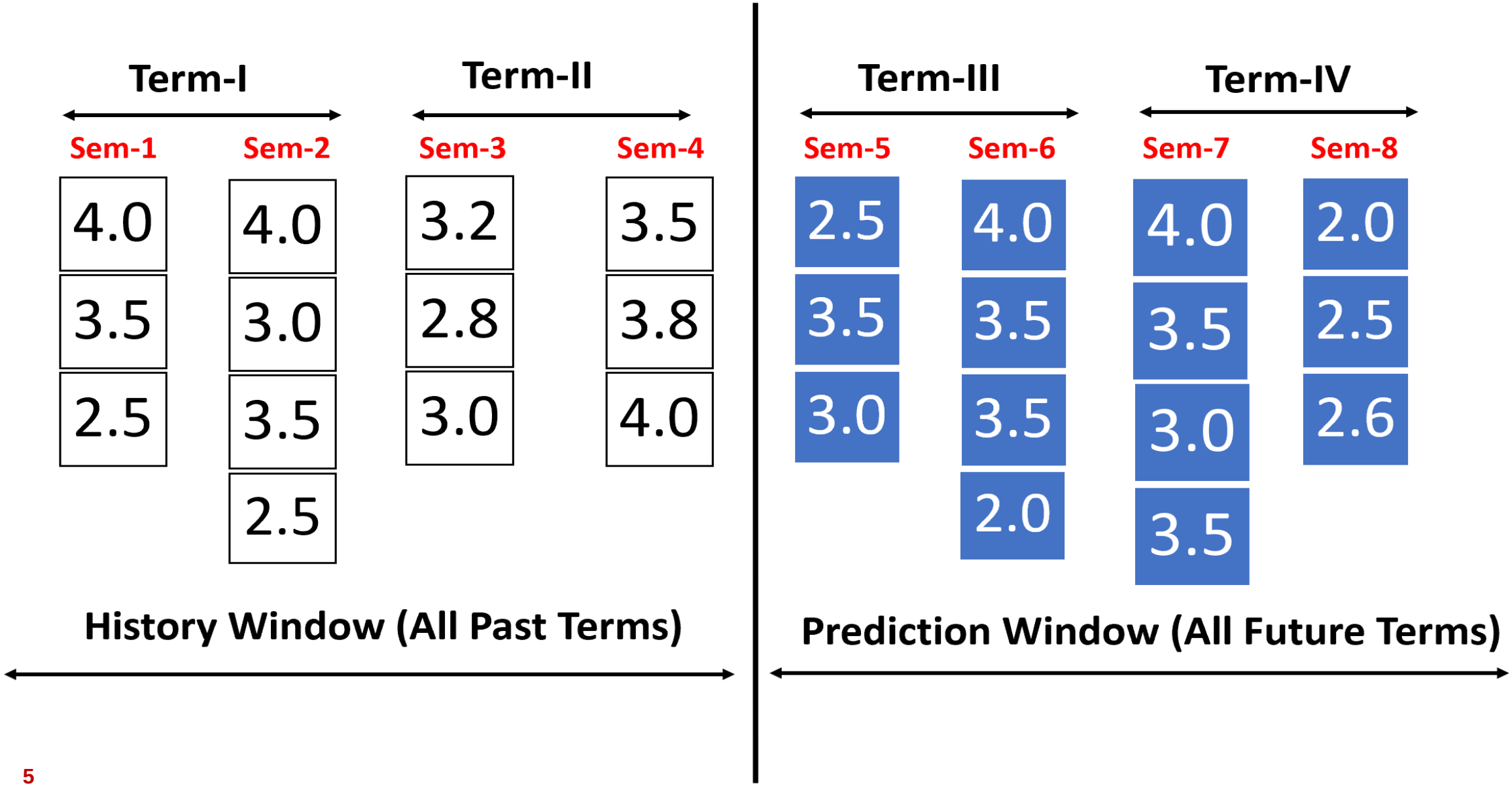
- **Data:** 9-years of **Undergraduate data** containing 1K+ student records is used for this study.
- **Key Features:** Core Courses, Sub Courses, Credit hours, Course Code, Grades, Core Courses GPA, Student Term
- **Core Courses** taken by the students are as follows:

Term	Core Courses List
First Year	Classical/Modern Foreign Language Sequence, Composition, Computational Reasoning, Philosophy of Human Nature, Faith and Critical Reason
Sophomore Year	Natural Science, Physical Science, Life Science, Literature, History, Social Science, Fine ad Performing Arts
Junior Year	Philosophical Ethics, Theology
Senior Year	Interdisciplinary Capstone, Seminar,



Grade	A	A-	B+	B	B-	C+	C-	D	F
Points	4.00	3.67	3.33	3.00	2.67	2.33	2.00	1.00	0.00

Sample Student Data



Research Questions

	Data Imputation	Final GPA Prediction	Analyzing Historical Data	Model Implementation	Best Model
Outline					
Question	**Q.1. How did you impute the missing data? Did you just drop it?**	**Q.2. What terms to use for final grade point GPA prediction?**	**Q.3. What are the key attributes of historical data and how to analyze it ?**	**Q.4 How to apply the generated algorithm in the prediction window?**	**Q.5 What parameter is used to identify the best model?**
Exp. Design	• *Combined the core courses as one course* • *Drop the missing rows* • *Replace the missing values by taking the mean or median of the data.*	• *All the course terms – First, Second, Third and Fourth were used to determine the final grade point average of the student.*	• *Various python-based data mining and machine learning algorithms were developed to understand the relationship between term grades and Final GPA.* • *Key terms used are – Core Courses, first term GPA, second term GPA, course code, and final GPA.*	• *The generated algorithm were applied to determine the GPA of third and fourth term.* • *Additionally, term wise plots has been created to understand the performance of the student in each term.*	• *Accuracy plots were used to find the best model.* • *MSE, Coefficient of Determination and MEAN RMSE are some other parameters which were used to compare the different models.*
Outcomes	**Data Imputed**	**Student's Final GPA**	**First Term, Second Term and Final GPA**	**Predicted Third and Fourth Term GPA**	**Model Comparison and Evaluation**

Step 1: Data Imputation and Final GPA Prediction

Raw Dataset

	SI	Programs	Code	CPrograms	Level	Scores
0	RA01	English	1102	COMPOSITION II	first term	0.00000
1	RA01	Mathematics	1206	CALCULUS I	first term	1.33333
2	RA01	Philosophy	1000	PHIL OF HUMAN NATURE	first term	1.33333
3	RA01	Theology	1000	FAITH & CRITICAL REASON	first term	2.13667
4	RA01	English	1102	COMPOSITION II	first term	2.53333

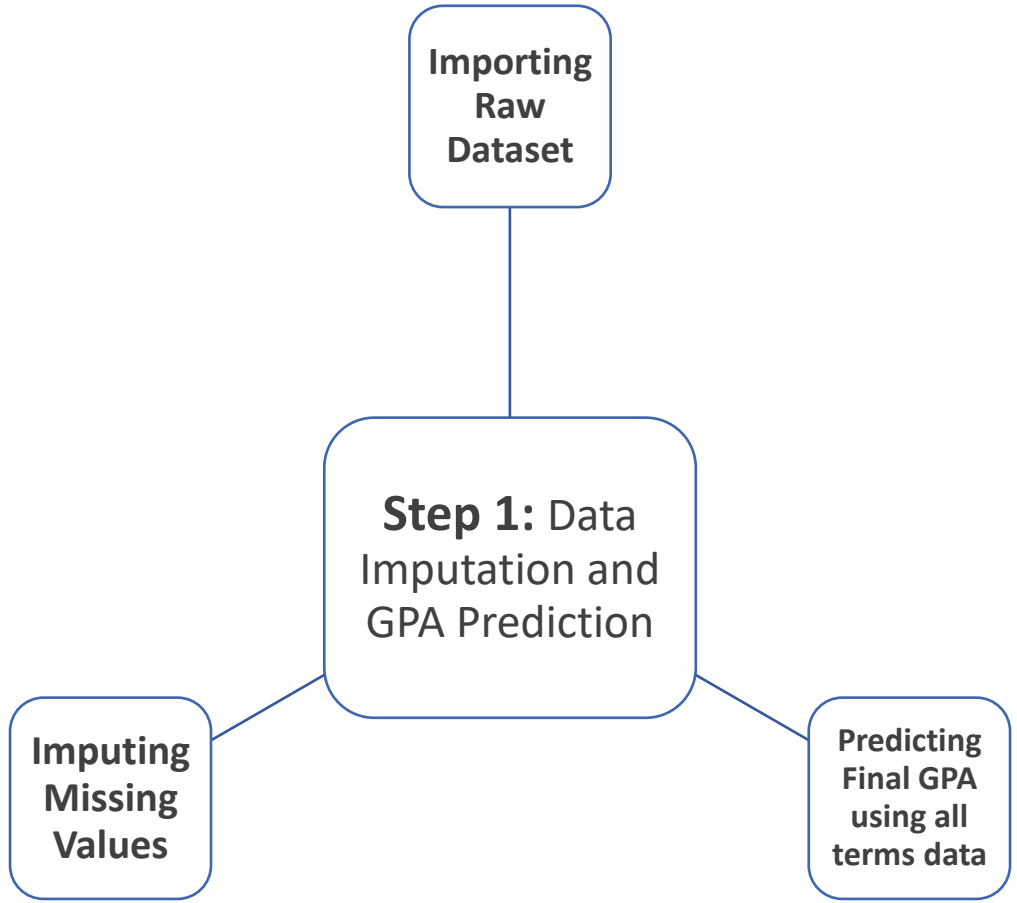
Imputed Dataset

Level	SI	first term	fourth term	second term	third term
0	RA01	1.687777	2.532222	2.386666	1.571667
1	RA02	1.814666	2.737336	2.469584	1.825000
2	RA03	1.344666	0.566667	3.217143	3.518335
3	RA04	1.669443	3.388094	2.916666	2.472381
4	RA05	1.595000	1.820557	1.741667	3.071667
...	...	...	...	...	...

Final GPA Prediction

Level	SI	first term	fourth term	second term	third term	grad_GPA
0	RA01	1.687777	2.532222	2.386666	1.571667	2.044583
1	RA02	1.814666	2.737336	2.469584	1.825000	2.211646
2	RA03	1.344666	0.566667	3.217143	3.518335	2.161703
3	RA04	1.669443	3.388094	2.916666	2.472381	2.611646
4	RA05	1.595000	1.820557	1.741667	3.071667	2.057222
..	...	...	...	...	...	...

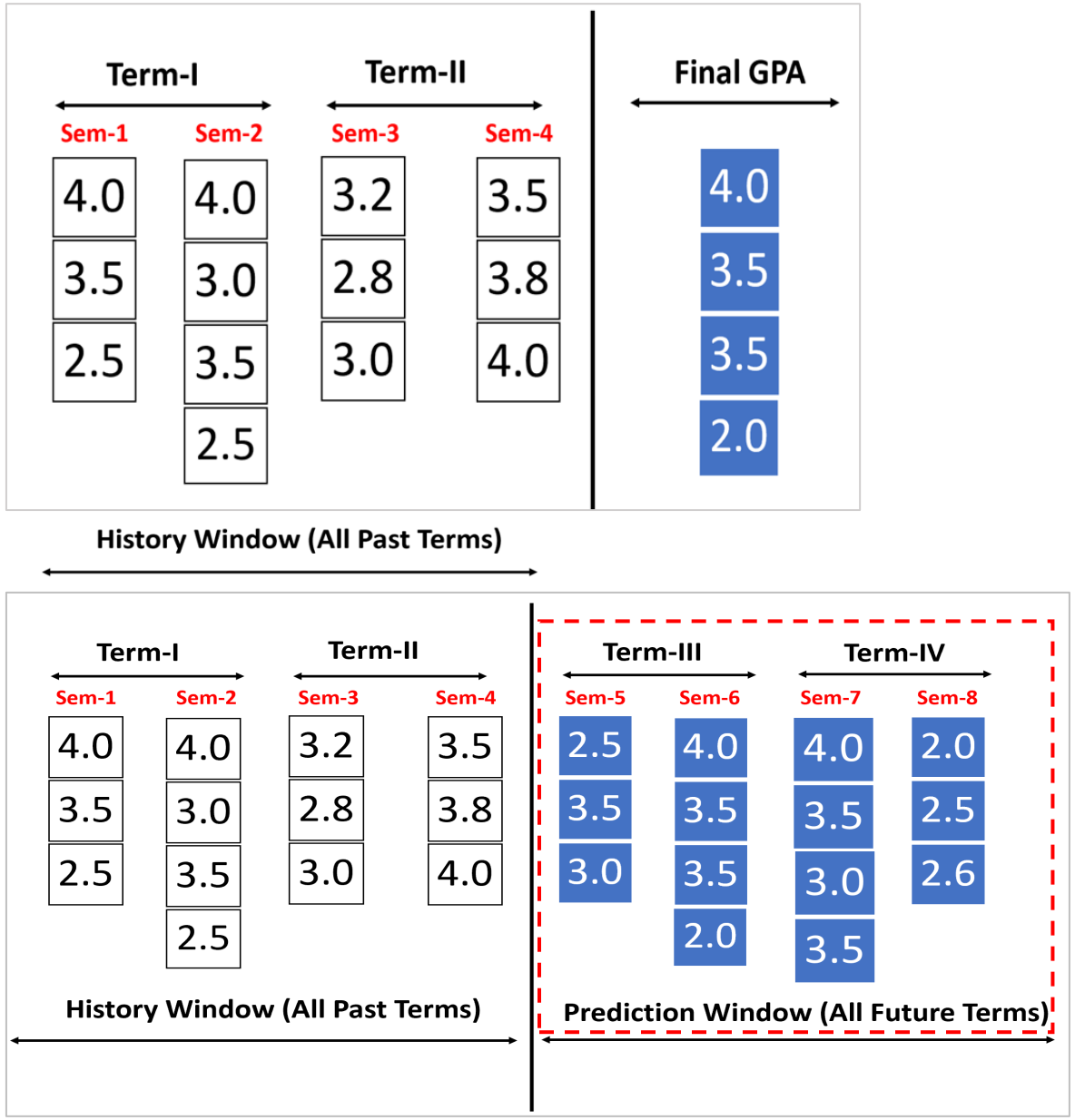
General Idea



Python Libraries

certifi==2020.11.8; contextlib2==0.6.0.post1; Cython==0.29.20; joblib==0.17.0; lxml==4.5.1; numpy==1.19.4; pandas==1.1.4 Pillow==7.1.2; python-dateutil==2.8.1; pytz==2020.4; scikit-learn==0.23.2

Step 2: Entering Prediction Window



Using the history window, what do you want to find in the Prediction Window?

1) What will be the predicted Third Year GPA?

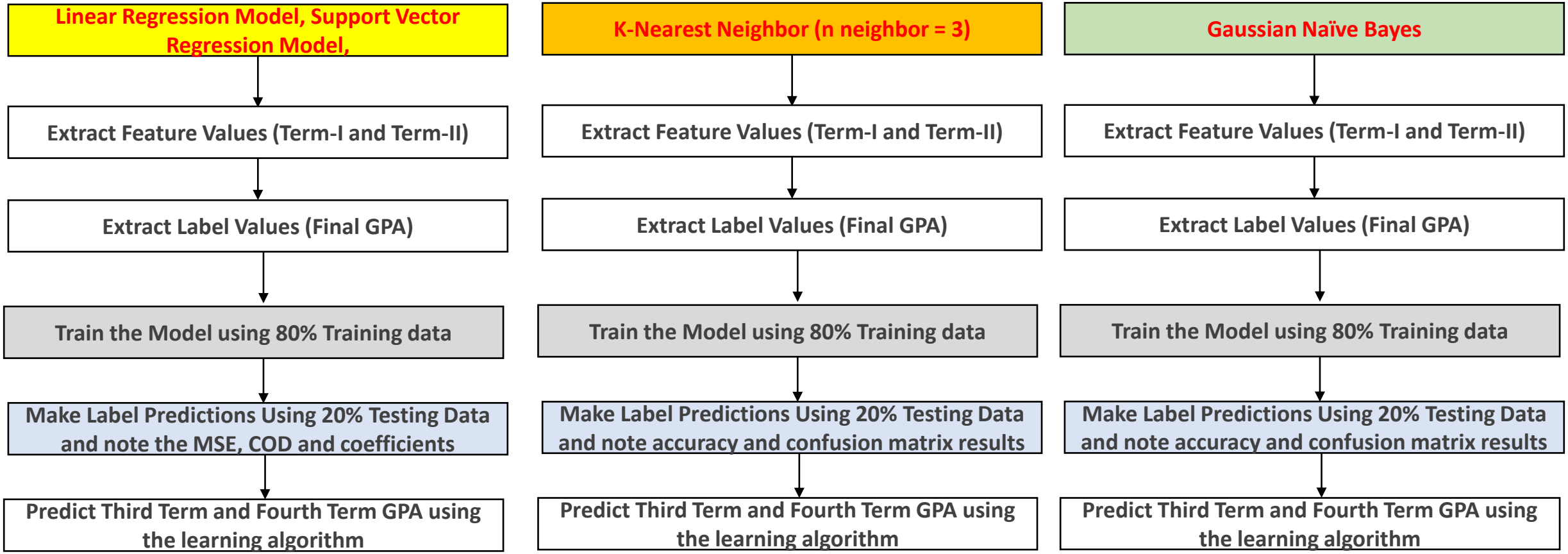
2) What will be the predicted Fourth Year GPA?

3) What will be the Cumulative GPA of all terms?

4) What will be the MSE or Accuracy of models used?

5) Which model is best to work on prediction window?

Step 3: Python-based Data Mining / ML Models



Regression Code

```
from sklearn.model_selection import train_test_split
import numpy as np
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score
#extact the feature values
features = ['first term']

X = final_dataset.loc[:, features].values
#extract the label values
y = final_dataset['grad_GPA']

#define train and test dataset of X and y respectively
X_train, X_test, y_train, y_test = train_test_split (X, y, test_size=0.2, random_state=0)

# Create linear regression object
regr = linear_model.LinearRegression()

# Train the model using the training sets
regr.fit(X_train, y_train)

# Make predictions using the testing set
y_pred = regr.predict(X_test)

# The coefficients
print('Coefficients: \n', regr.coef_)
# The mean squared error
print('Mean squared error: %.2f'
      % mean_squared_error(y_test, y_pred))
# The coefficient of determination: 1 is perfect prediction
print('Coefficient of determination: %.2f'
      % r2_score(y_test, y_pred))

print(y_pred)
```

```
new_feature1 = df1[['third term']]
new_feature2 = df1[['fourth term']]
new_pred1 = regr.predict(new_feature1)
new_pred2 = regr.predict(new_feature2)
print(new_pred1)
print(new_pred2)

#plotting
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 10))
plt.plot(new_pred1)
plt.xlabel('number of students')
plt.ylabel('Third Term GPA')
plt.title('third term Results')


plt.figure(figsize=(10, 10))
plt.plot(new_pred2)
plt.xlabel('number of students')
plt.ylabel('Fourth Term GPA')
plt.title('fourth term Results')


plt.show()
```

Regression Code

```
from sklearn.model_selection import train_test_split
import numpy as np
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score
#extact the feature values
features = ['second term']

X = final_dataset.loc[:, features].values
#extract the label values
y = final_dataset['grad_GPA']

#define train and test dataset of X and y respectively
X_train, X_test, y_train, y_test = train_test_split (X, y, test_size=0.2, random_state=0)

# Create linear regression object
regr1 = linear_model.LinearRegression()

# Train the model using the training sets
regr1.fit(X_train, y_train)

# Make predictions using the testing set
y_pred = regr1.predict(X_test)

# The coefficients
print('Coefficients: \n', regr1.coef_)
# The mean squared error
print('Mean squared error: %.2f'
      % mean_squared_error(y_test, y_pred))
# The coefficient of determination: 1 is perfect prediction
print('Coefficient of determination: %.2f'
      % r2_score(y_test, y_pred))

print(y_pred)
```

```
Coefficients:
 [0.1510986]
Mean squared error: 0.02
```

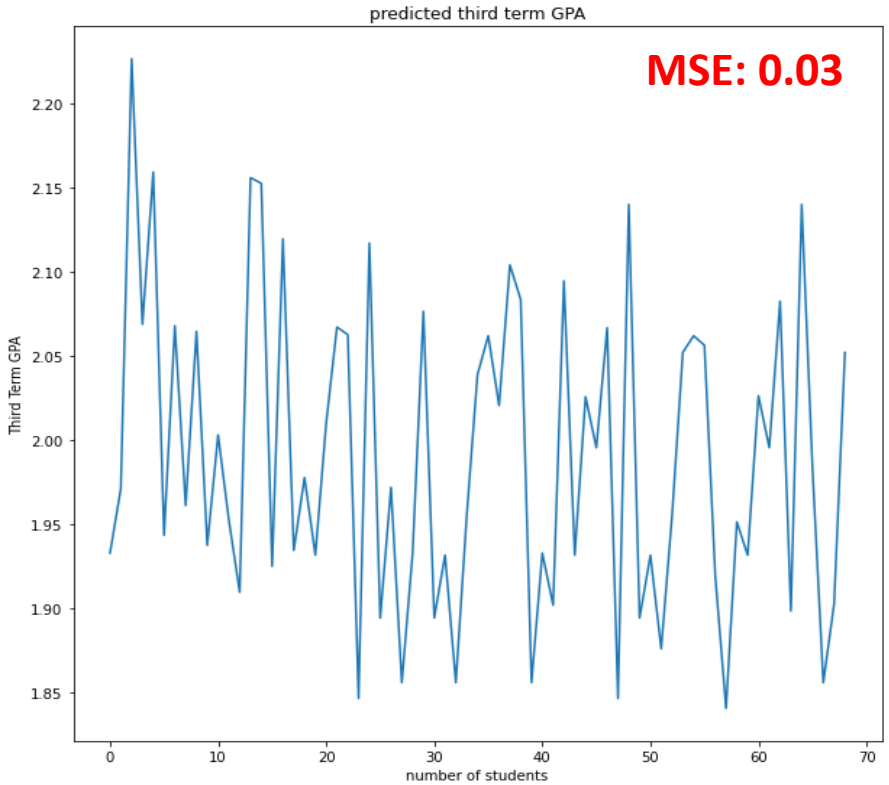
```
new_feature3 = df1[['third term']]
new_feature4 = df1[['fourth term']]
new_pred3 = regr1.predict(new_feature3)
new_pred4 = regr1.predict(new_feature4)
print(new_pred3)
print(new_pred3)

#plotting
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 10))
plt.plot(new_pred3)
plt.xlabel('number of students')
plt.ylabel('Third Term GPA')
plt.title('predicted third term GPA')


plt.figure(figsize=(10, 10))
plt.plot(new_pred4)
plt.xlabel('number of students')
plt.ylabel('Fourth Term GPA')
plt.title('predicted fourth term GPA')


plt.show()
```

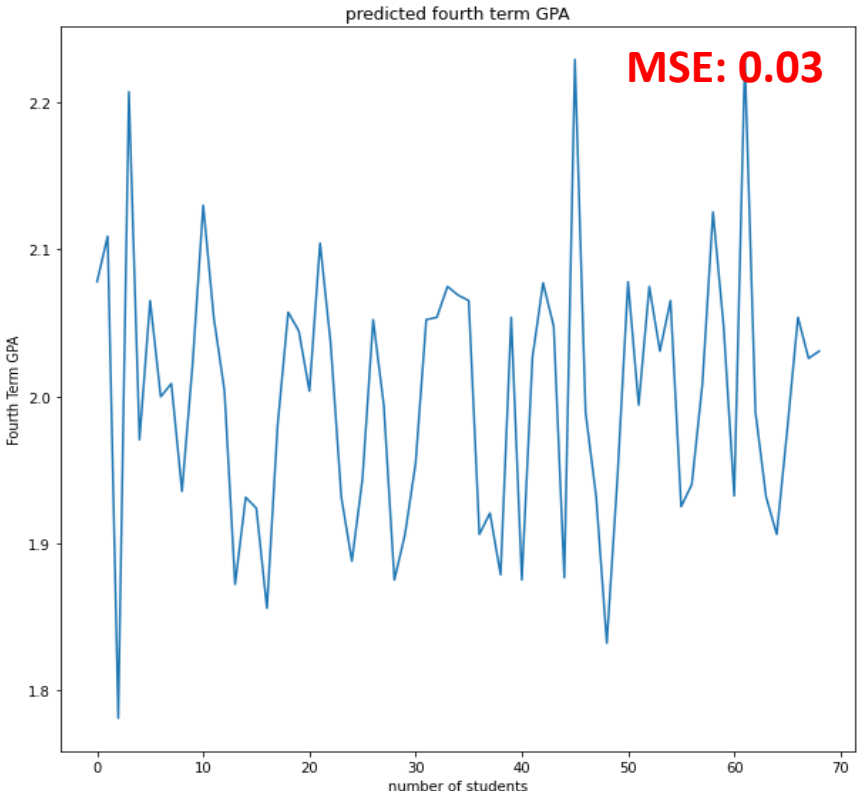
GPA Results

GPA Results of All Models

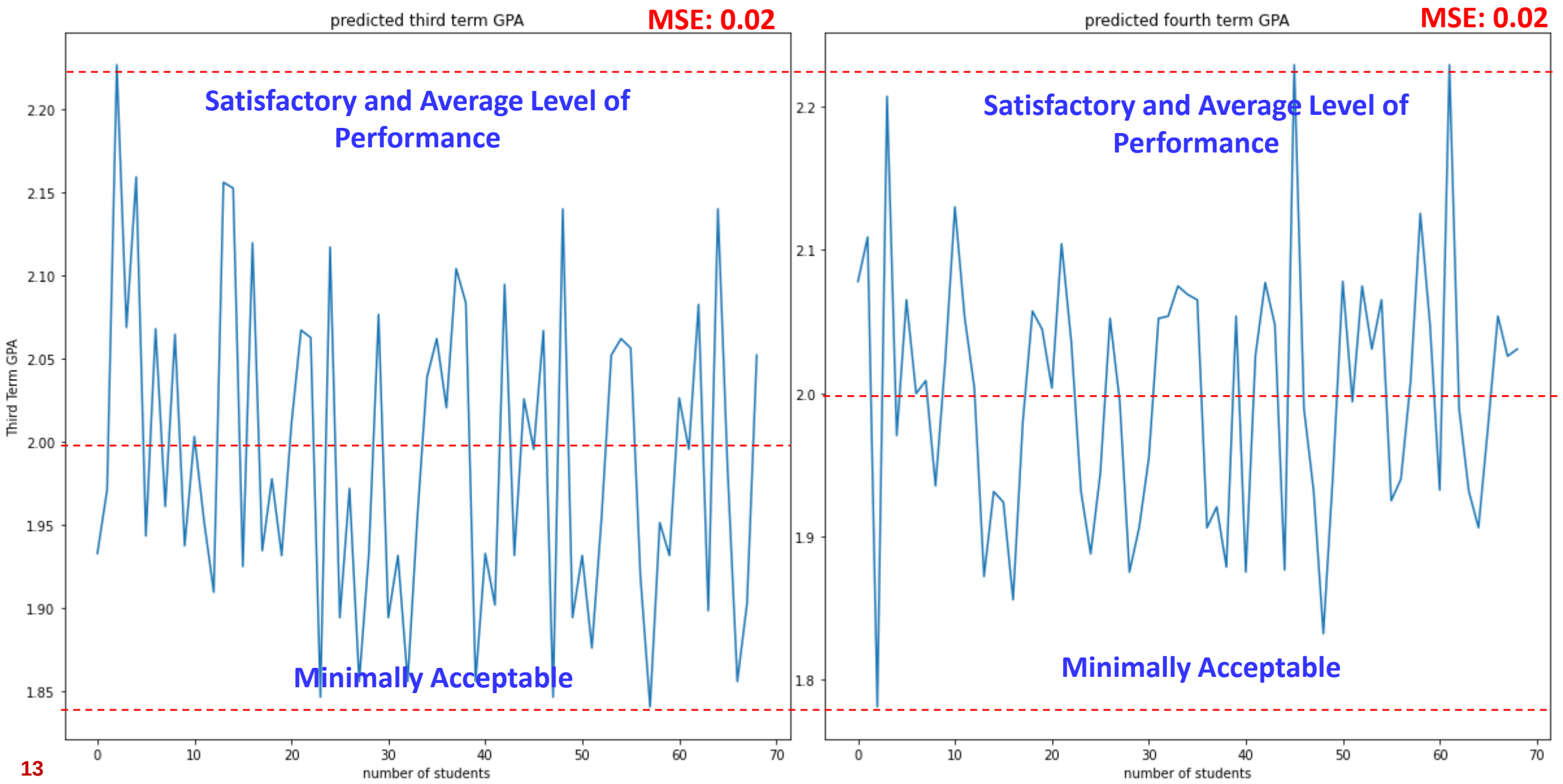
Student Data	Student-1				Student-2				Student-3			
Model	Term-I	Term-II	Term-III	Term-IV	Term-I	Term-II	Term-III	Term-IV	Term-I	Term-II	Term-III	Term-IV
Linear Regression	1.68	2.38	1.93	2.07	1.81	2.46	1.96	2.1	1.34	3.21	2.21	1.78
SVR	1.68	2.38	1.95	2.05	1.81	2.46	1.9	2.18	1.34	3.21	2.23	1.6
KNN	1.68	2.38	1	2	1.81	2.46	2	2	1.34	3.21	2	1
Naïve Bayes	1.68	2.38	1	2	1.81	2.46	1	2	1.34	3.21	2	1

SVR Predictions





Linear Regression Predictions



K-Nearest Neighbor

Class Labeling

```
label = []

for num in final_dataset['grad_GPA']:
  if num < 2.00:
    label.append(1)
  elif num >= 2.00 and num < 3.00:
    label.append(2)
  else:
    label.append(3)
```

Level	SI	first term	second term	grad_GPA	label
0	RA01	1.687777	2.386666	2.044583	2
1	RA02	1.814666	2.469584	2.211646	2
2	RA03	1.344666	3.217143	2.161703	2
3	RA04	1.669443	2.916666	2.611646	2
4	RA05	1.595000	1.741667	2.057222	2
...	...	...	...	...	...
64	RA66	2.682220	1.986665	2.251944	2
65	RA67	2.271211	1.632820	1.919416	1

Class 1: Minimally Acceptable

Class 2: Satisfactory or Average Level of Performance

K-Nearest Neighbor and NB

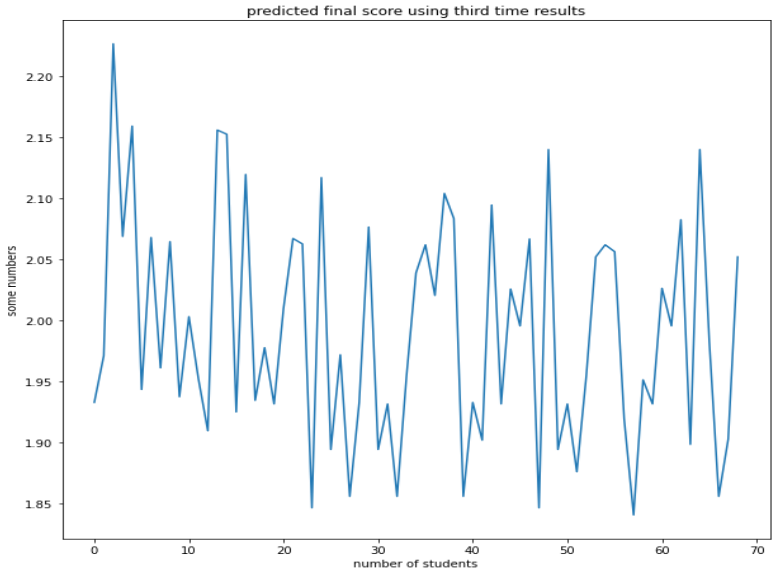
KNN

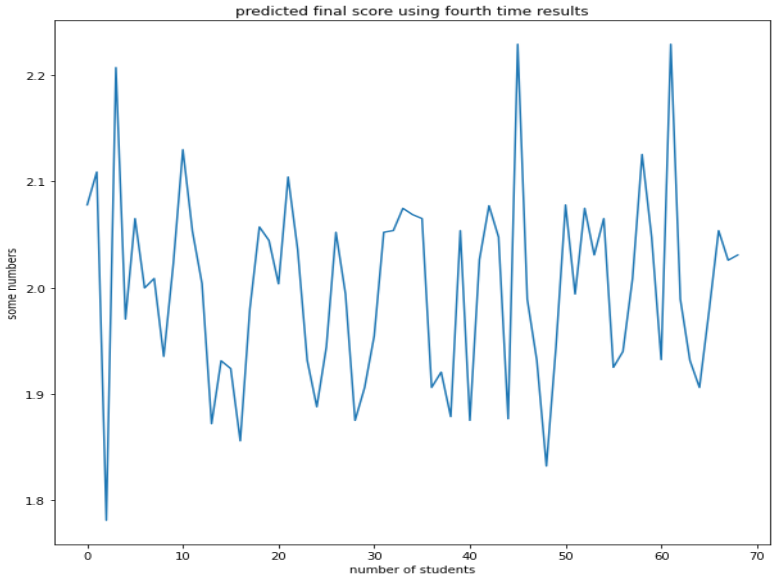
```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix
import numpy as np
#split feature set from the data frame
features = ['first term','second term']
#split the train and test data
X = final_dataset.loc[:, features].values
y = final_dataset['label']


X_train, X_test, y_train, y_test = train_test_split (X, y, test_size=0.2, random_state=0)

# Create linear regression object
clf = KNeighborsClassifier(n_neighbors=3)

# Train the model using the training sets
clf.fit(X_train, y_train)

# Make predictions using the testing set
y_pred = clf.predict(X_test)


%time
from sklearn.metrics import classification_report
y_pred = clf.predict(X_test)

print('accuracy %s' % accuracy_score(y_pred, y_test))
print(classification_report(y_test, y_pred))
```

NB

```
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix
import numpy as np

features = ['first term','second term']

X = final_dataset.loc[:, features].values
y = final_dataset['label']


X_train, X_test, y_train, y_test = train_test_split (X, y, test_size=0.2, random_state=0)

# Create linear regression object
nbc = GaussianNB()

# Train the model using the training sets
nbc.fit(X_train, y_train)

# Make predictions using the testing set
y_pred = nbc.predict(X_test)

%time
from sklearn.metrics import classification_report
y_pred = nbc.predict(X_test)

print('accuracy %s' % accuracy_score(y_pred, y_test))
print(classification_report(y_test, y_pred))
```

Accuracy Results





Accuracy Results

Method	Term-I	Term-II	Cumulative (Term-I and Term-II)
KNN	0.42	0.78	0.57
NB	**0.78**	**0.78**	**0.65**

Key Observations:

- The Best performance by the end of Term-I , is the NB method with term performance represented by course GPA, 78%.
- When the prediction is performed later the best performance is systematically obtained again with the NB Methods, but with the term prediction as 65%
- It is always better to do the predictions by the end of Term II

Conclusions

- Future grade prediction of undergraduate students
- Developed python-based algorithms to predict the future Term GPA using historical term data.
- Linear regression model showed low MSE and helped in identifying the low-GPA students.
- Further, KNN and Naïve Bayes models helped in finding the best term for prediction. In this case study, Term-II is found to be the best term for making predictions.

Usefulness of the Study

- Enable the education institution to identify not to complete the program.
- Identify students at risk and provide adequate advising and tailored help towards reduced failure rates.
- Detecting high and low achiever students and help them enhance their career paths.
- Predictors and Early Warning Systems can be developed
- Risk of Failing a Course
- Drop out Risk
- Grade Prediction
- Graduation Rate

Selected References

- Hu, Q., & Rangwala, H. (2019, March). Reliable deep grade prediction with uncertainty estimation. In Proceedings of the 9th International Conference on Learning Analytics & Knowledge (pp. 76-85).
- Hu, Q., Polyzou, A., Karypis, G., & Rangwala, H. (2017, October). Enriching course-specific regression models with content features for grade prediction. In 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA) (pp. 504-513). IEEE.
- Kumar, M., & Singh, A. J. (2017). Evaluation of Data Mining Techniques for Predicting Student's Performance. International Journal of Modern Education & Computer Science, 9(8).
- Patil, A. P., Ganesan, K., & Kanavalli, A. (2017, December). Effective deep learning model to predict student grade point averages. In 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC) (pp. 1-6). IEEE.
- Okubo, F., Yamashita, T., Shimada, A., & Konomi, S. (2017, January). Students' performance prediction using data of multiple courses by recurrent neural network. In 25th International Conference on Computers in Education, ICCE 2017 (pp. 439-444). Asia-Pacific Society for Computers in Education.
- Hunt, M., Lin, S., & Kulkarni, C. Predicting Course Grades.
- Al-Barrak, M. A., & Al-Razgan, M. (2016). Predicting students final GPA using decision trees: a case study. International journal of information and education technology, 6(7), 528.
- Elbadrawy, A., Polyzou, A., Ren, Z., Sweeney, M., Karypis, G., & Rangwala, H. (2016). Predicting student performance using personalized analytics. Computer, 49(4), 61-69.
- Polyzou, A., & Karypis, G. (2016, April). Grade prediction with course and student specific models. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 89-101). Springer, Cham.

Summary of Reviewed Work

Year	Title of the Paper	Recommendation System	Data Source	Features Used	Task or Application
2019	Reliable deep grade prediction with uncertainity estimation	Bayseian deep learning models - (1) Multilayer perception (MLP) (2) Long short term memory (LSTM) networks	Large five undergraduate majors (including computer science, electrical engineering, biology, psychology, civil engineering) were chosen. To build a course-specific model for a target course, prior courses according to the University Catalog from Fall 2009 to Spring 2017 were chosen.	Number of students, number of courses, number of grades	Course-specific Bayesian deep learning models for grade prediction namely MLP and LSTM were proposed. The proposed models can provide prediction uncertainity (essential for decision making) and can also identify influential courses that result in a student's failure of a course. The authors further evaluated the models' capability of catching at-risk students.
2017	Enriching course-specific regression models with content features for grade prediction	Linear Regression and Matrix Factorization	Data set obtained from four departments at George Mason University. The data (required courses and electives) was collected from Fall 2009 to Spring 2016. *Removed courses whose grades were pass/fail	(1) Student related features include their demographic data (such as age, gender, high school GPA), GPA of student as of last term, student's academic level. (2) Course related features include discipline, credit hours, course level, course difficulty information (GPA is the indicator for the same) (3) Instructor related features include rank, tenure status, and the GPA of courses he/she has taught	The authors proposed a hybrid course-specific regression model enriched with features about students, courses and instructors. It was found that incorporating content features can boost the perfomance of the model. For degree programs with high flexibility , predicting the grades with only content feature can give better results.
2017	Evaluation of Data Mining Techniques for Predicting Student's Performance	Decision Tree, Naïve Bayes, Random Forest, PART and Bayes Network with three most important techniques such as 10-fold cross-validation, percentage split (74%) and training set.	Student's data of 412 post-graduates is taken	Student Attributes: Age, Gender, Home Location, Communication skill, Sportsperson, social friends, smoking habits, drinking habits, interest in study, internet and hosteller, day scholar Academic Attributes: 10th %age, 10th board, 10th education medium, 12th %age, 12th board, 12th education medium JEE Rank, Admission type, institution type etc.	Best model for predicting student's performance is identified. It was found that Random forest algorithm showed the best result as compared to another algorithms mainly Decision Tree, Naïve Bayes, Bayes Network and CART. The correctly classified instance with Random Forest algorithm is 61.40% with a very good recall value equal to 1.

Summary of Reviewed Work

Year	Title of the Paper	Recommendation System	Data Source	Features Used	Task or Application
2017	Students' performance prediction using data of multiple courses by recurrent neural network	Recurrent Neural Network	Learning logs from 937 students who attended one of six courses by two teachers were collected		Showed a method to predict student's final grade using recurrent neural network. Important learning activities for obtaining a specific grade by observing the values of weight of the trained RNN were also identified.
2016	Grade Prediction with Course and Student Specific Models	Sparse linear models and low rank matrix factorization	Student-course-grade data set is obtained from the University of Minnesota which has a very flexible degree program.		The study presented two course-specific approaches based on linear regression and matrix factorization that perform better than existing approaches. Course specific subset of the data resulted in more accurate predictions.
2016	Predicting Course Grades	(1) Support Vector Machines (SVMs) and (2) Collaborative Filtering (CF) - Cosine Similarity and Pearson's Correlation	Stanford's CourseRank website (10, 000 anonymized student transcripts and around 7000 courses) data	Student's previous course grades, Recent GPA by department (last three quarters), Major, Concurrent courses, Planned weekly workload, No. of Courses previously taken	Construct one potential value-predictor - an estimated grade that a student will receive for a given course.
2016	Predicting students final GPA using decision trees: a case study.	Weka Tool Kit (algorithms implemented in Java), Classification Technique-Decision Trees	Transcript data for female students graduated in 2012 were collected from the data base management system (of King Saud University)and the total number of students was 235 students.	Student name, Student ID, Final GPA, Semester of graduation, Major, Nationality, Campus, All the courses taken by the student including the course' grade	Used education data mining to predict students' final GPA based on their grades in previous courses. J48 algorithm was implemented to discover classification rules. Most important courses in the student's study plan based on their grades in the mandatory courses were also identified.
2016	Predicting student performance using personalized analytics	Multiregression and Matrix Factorization Techniques	Trained and tested performance prediction methods on four datasets: George mason university transcript data, University of minnesota (UMN) transcript data, UMN LMS data and Stanford University MOOC data.	Student and course number, course instructor, course level and the department offering that course	Forecast students' grades in future courses as well as in-class assessments.

Acknowledgements

- Dr. Gary Weiss, Department of Computer and Information Sciences
- Fordham University
- Education Data Mining Lab
- All my batch mates

Thank You