Predicting Final Exam Scores in Course 451 Based on quiz,homework,Presentation and Mid-Term Performance

RUIJING CHEN, JIREN LU, YITENG TU, ZEKAI XU, MINLIANG YU

Group 8

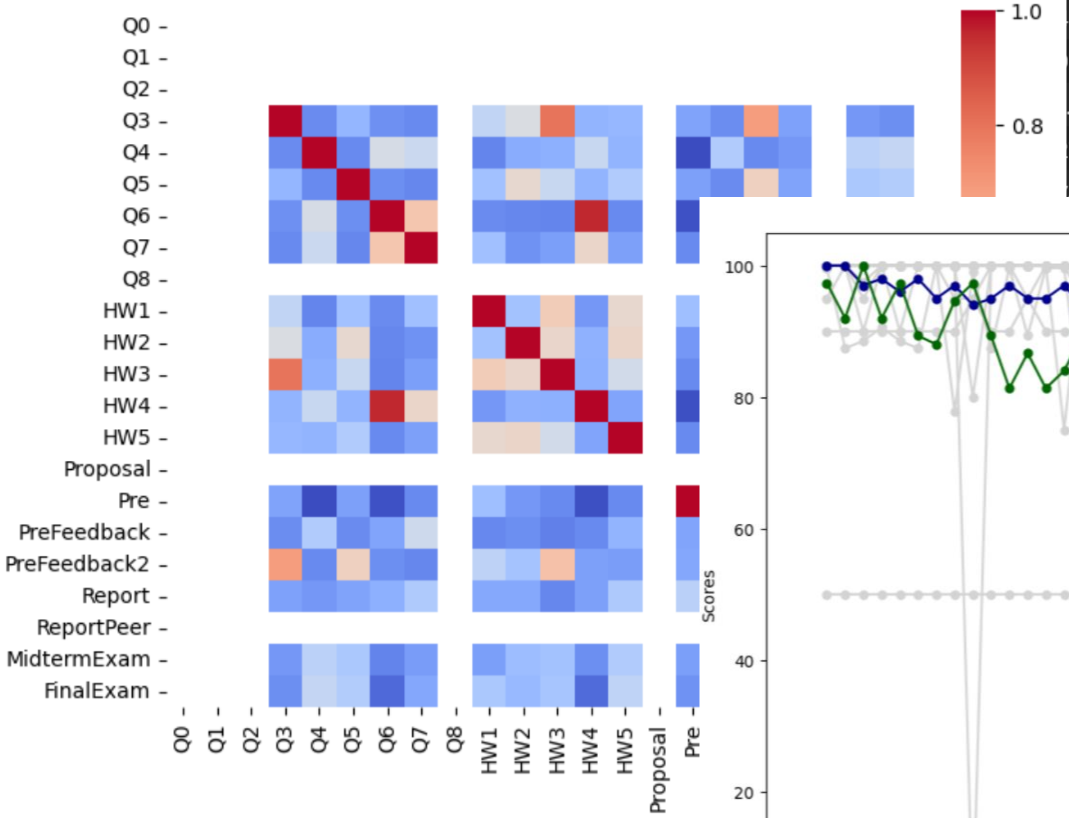
Our Task

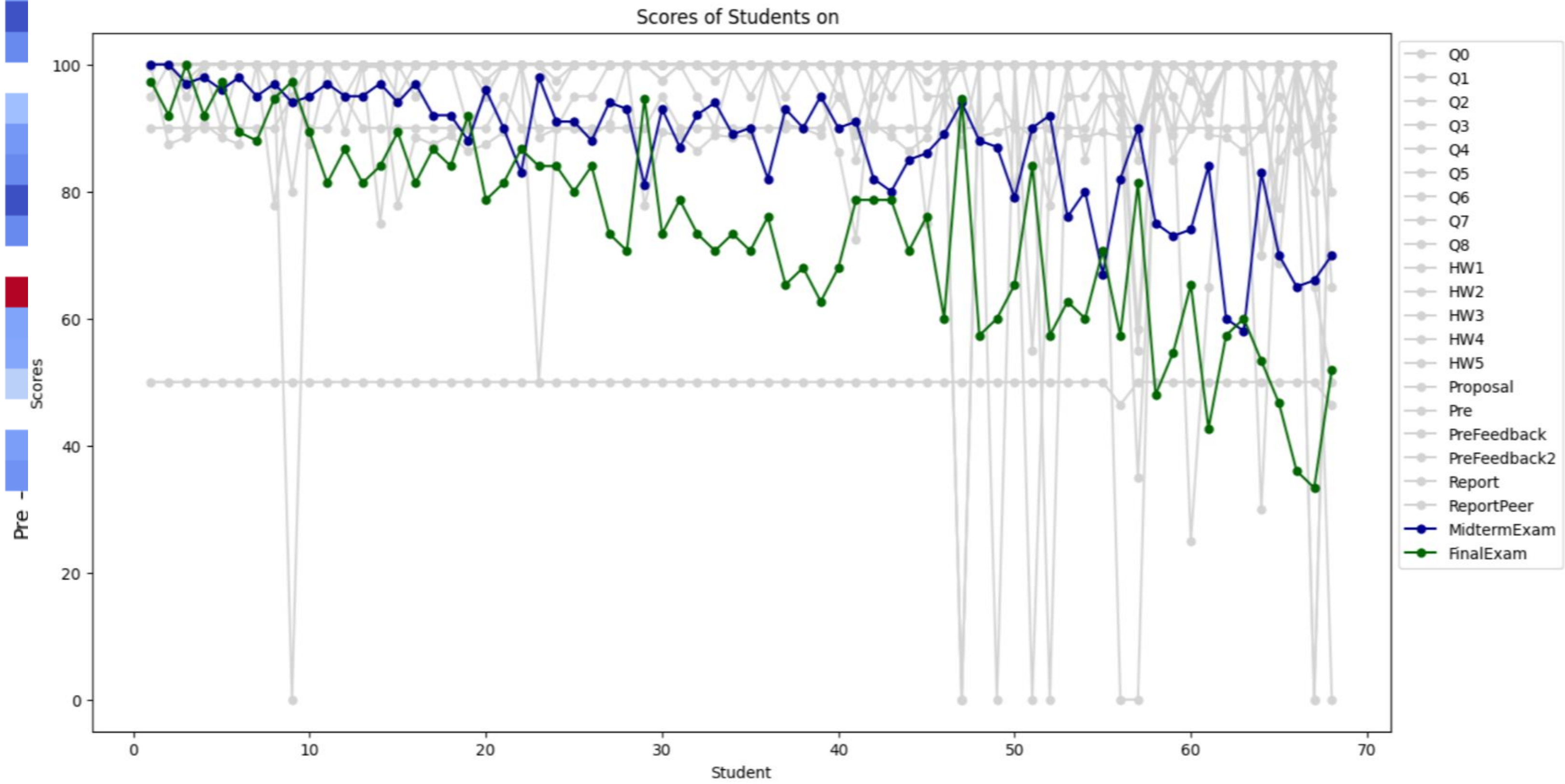
We aim to employ machine learning methods for predicting students' scores on the final exam based on their performance throughout the semester.

About Data Set

- Dataset Size: 68*22
- Dataset Quality: No missing values and outliers
- Features(scores of): Q1~Q8,HW1-HW5, Proposal, Pre,…,MidtermExam
- Target(scores of): FinalExam

	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	HW1	...	HW4	HW5	Proposal	Pre	PreFeedback	PreFeedback2	Report	ReportPeer	MidtermExam	FinalExam
1	4	7	5	12	16	14	10	3	4	20.0	...	20.0	20.0	7	26.90	2	2	9.0	4	100.0	73
2	4	7	5	12	16	14	10	3	4	20.0	...	20.0	20.0	7	23.60	2	2	9.0	4	100.0	69
3	4	7	5	12	16	14	10	3	4	19.0	...	20.0	18.0	7	23.91	2	2	9.0	4	97.0	75
															6	2	2	9.0	4	98.0	69
															1	2	2	9.0	4	96.0	73
															...	...	...	...	...	...	...
															7	2	2	9.0	4	83.0	40
															4	2	2	9.5	4	70.0	35





LASSO - variable selection

considering the multi-collinearity problem

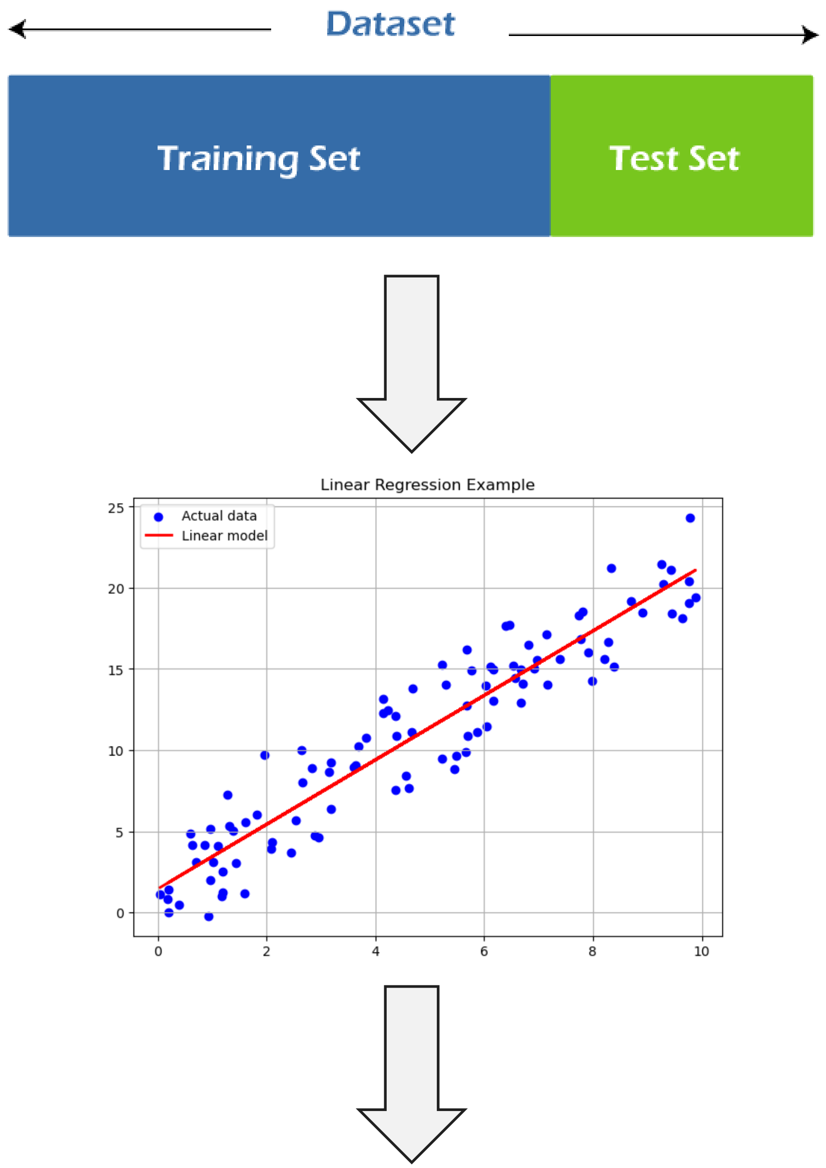
We choose:
Q3', 'Q4', 'Q6', 'Q7', 'HW1', 'HW2', 'HW3', 'HW4', 'HW5', 'Pre', 'PreFeedback', 'PreFeedback2', 'MidtermExam'

α	0.1	0.5	1.0	1.5	2.0
Q0	0.000000	0.000000	0.000000	0.000000	0.000000
Q1	0.000000	0.000000	0.000000	0.000000	0.000000
Q2	0.000000	0.000000	0.000000	0.000000	0.000000
Q3	-0.364910	-0.464571	-0.473636	-0.380286	-0.287303
Q4	0.507239	0.469236	0.433100	0.402644	0.372291
Q5	1.092193	0.401661	0.000000	0.000000	0.000000
Q6	-0.000000	-0.076210	-0.117167	-0.091232	-0.063635
Q7	-0.013912	-0.010848	-0.006689	-0.002999	-0.000000
Q8	0.000000	0.000000	0.000000	0.000000	0.000000
HW1	0.661138	0.621490	0.578999	0.539463	0.500621
HW2	-0.007557	0.005513	0.011186	0.007434	0.003764
HW3	0.099762	0.110337	0.103109	0.081319	0.059868
HW4	-0.192812	-0.108082	-0.060955	-0.082630	-0.105268
HW5	0.011330	0.014136	0.019882	0.028628	0.037286
Proposal	0.000000	0.000000	0.000000	0.000000	0.000000
Pre	-0.175653	-0.154666	-0.134154	-0.118243	-0.102221
PreFeedback	0.168505	0.170395	0.168828	0.163178	0.157781
PreFeedback2	0.016736	0.052426	0.068013	0.057746	0.047401
Report	-0.000000	-0.000000	-0.000000	-0.000000	-0.000000
ReportPeer	0.000000	0.000000	0.000000	0.000000	0.000000
MidtermExam	0.908393	0.914701	0.919628	0.922705	0.925748

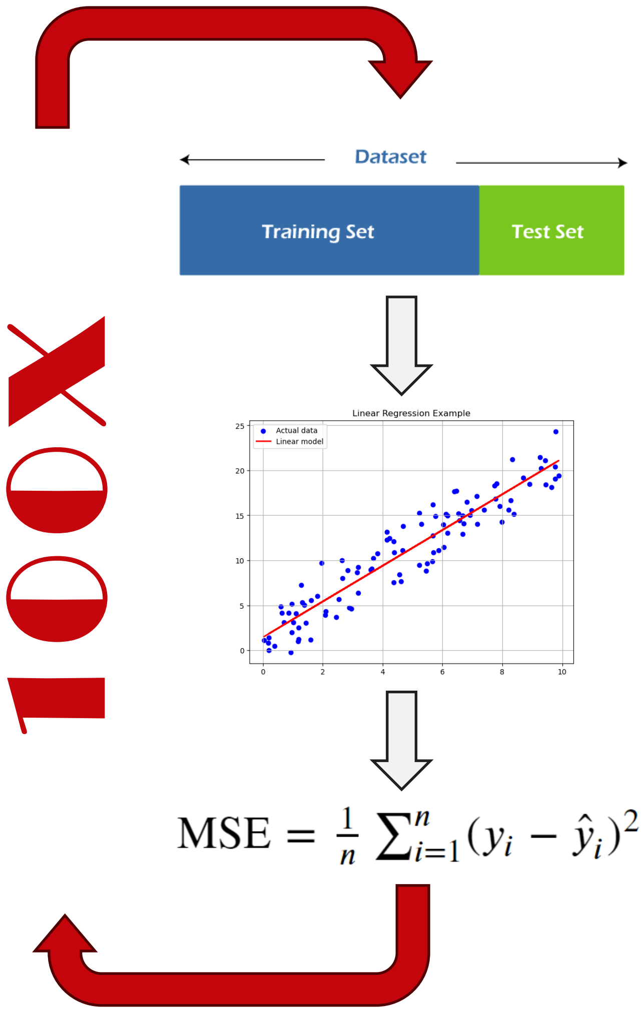
MSE (Mean Squared Error)

——Model Evolution Metrics

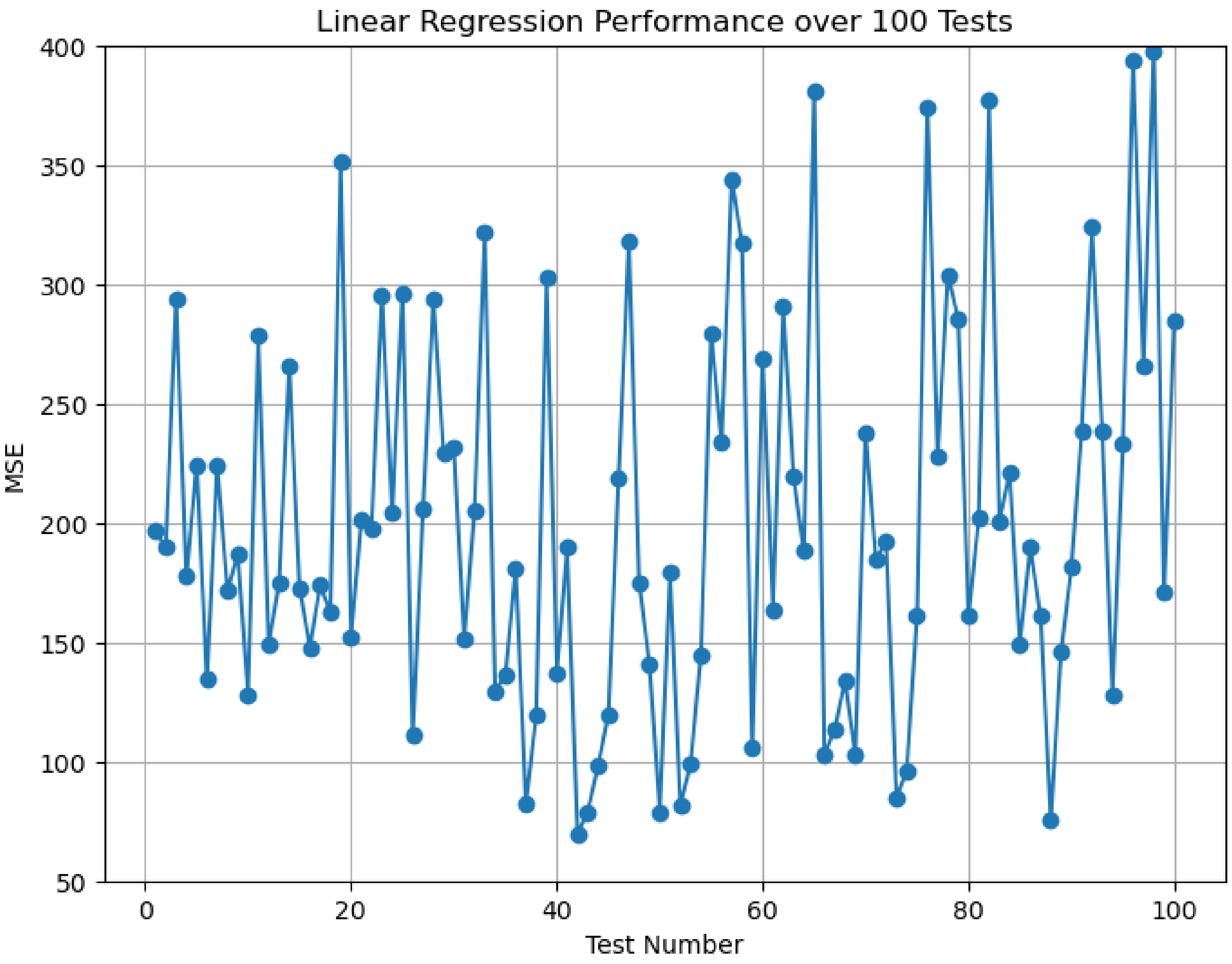
$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

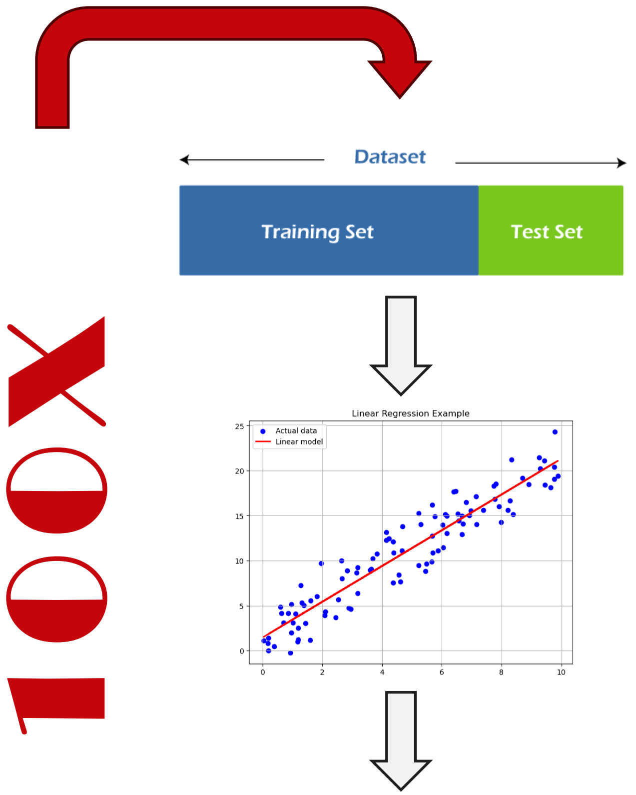


$$\mathrm{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

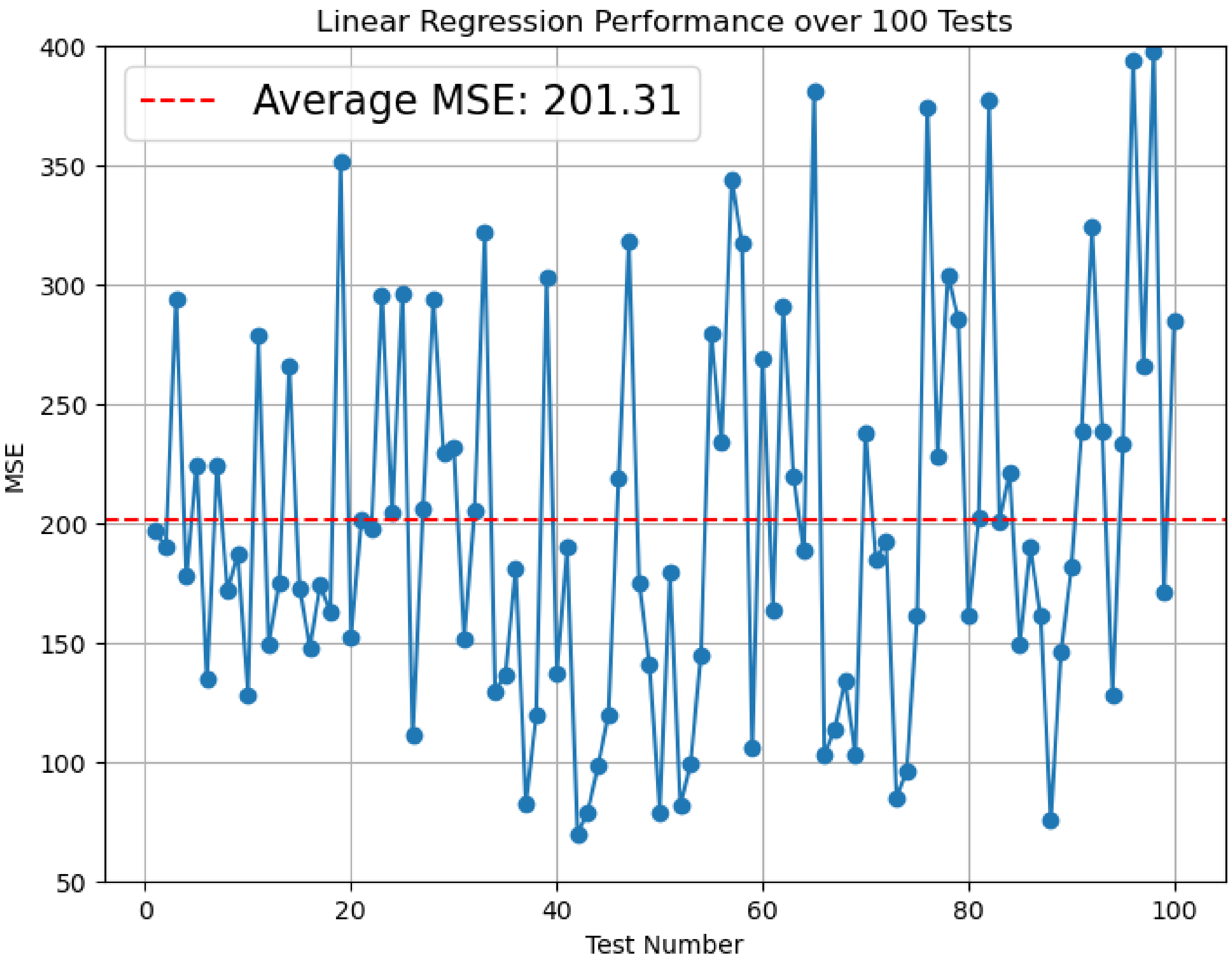


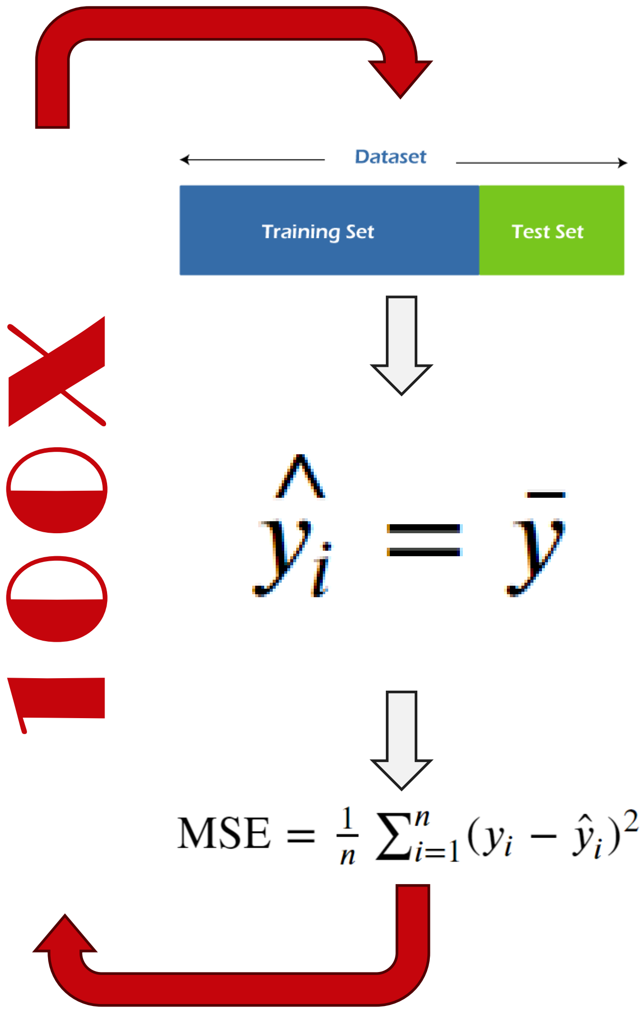
$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$





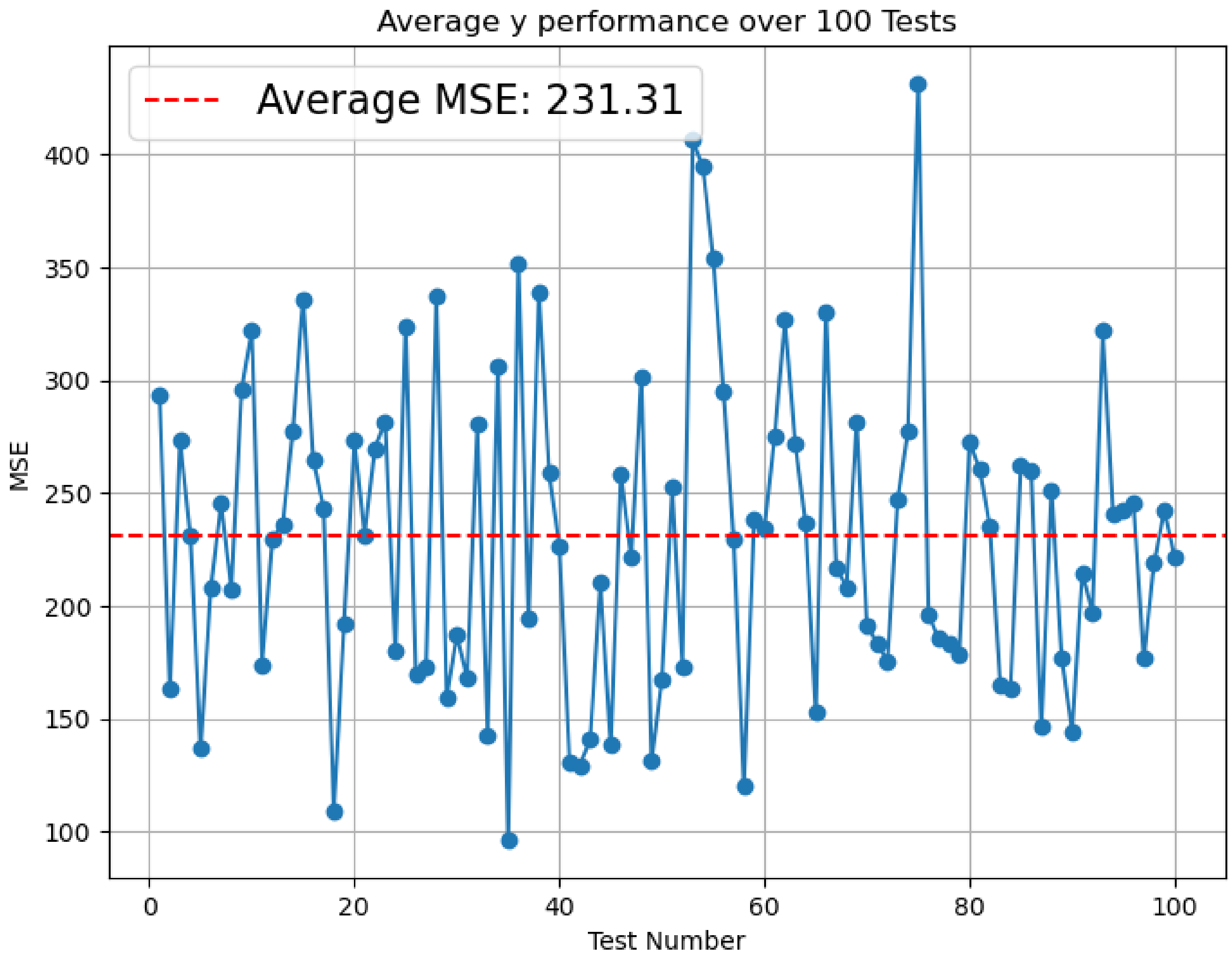
$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$





$$\hat{y}_i = \bar{y}$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$



GridSearchCV

A hyperparameter optimization selecting the best combination of model parameters from a predefined grid

Key parameters:

Estimator	The machine learning model to be used
param_grid	The names of model parameters + parameter values
cv	The number of folds
scoring	The metric used to evaluate model performance

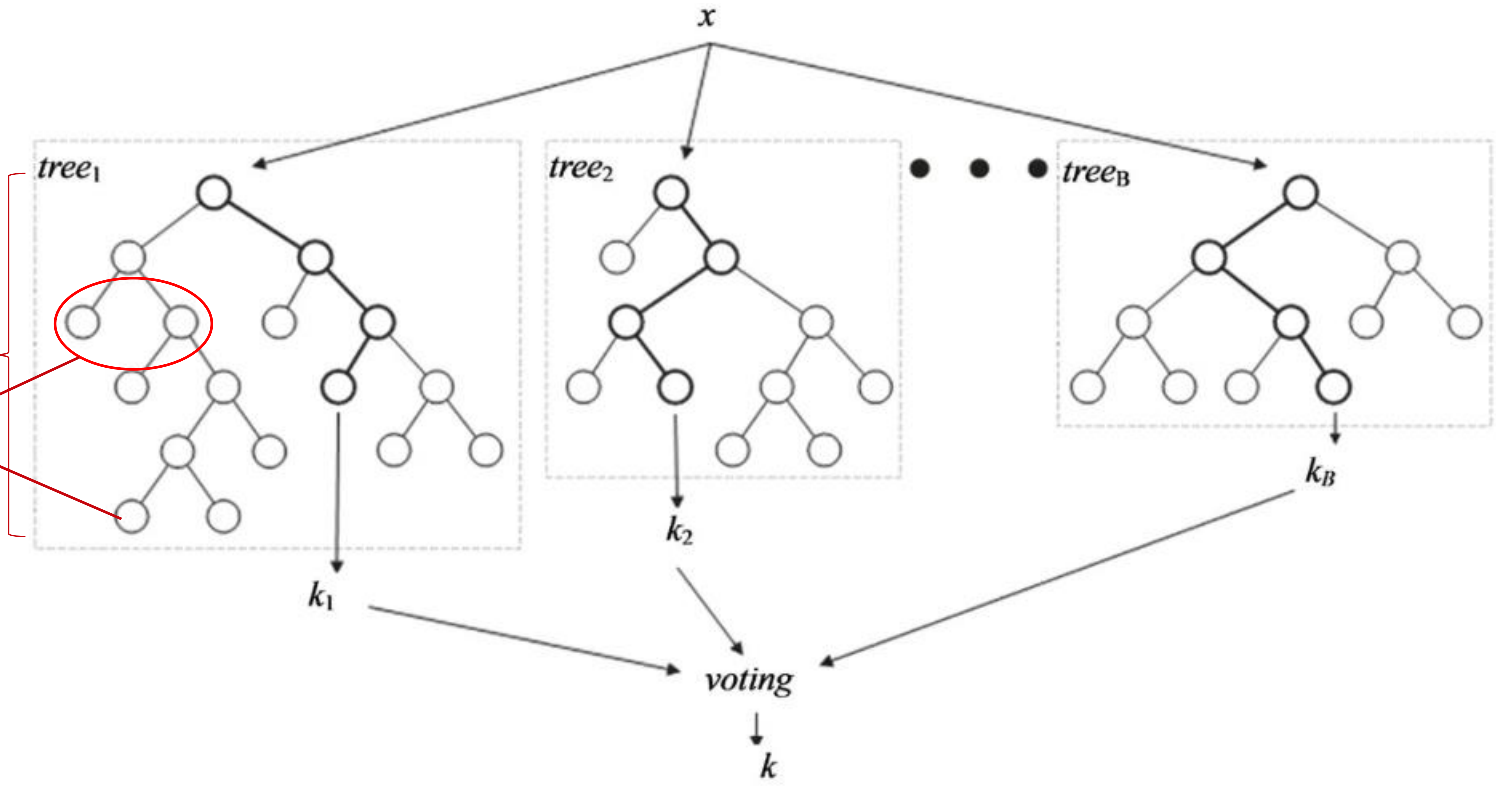
Random Forest

An ensemble learning method that constructs multiple decision trees during training

Key parameters:
max_depth

min_samples_leaf

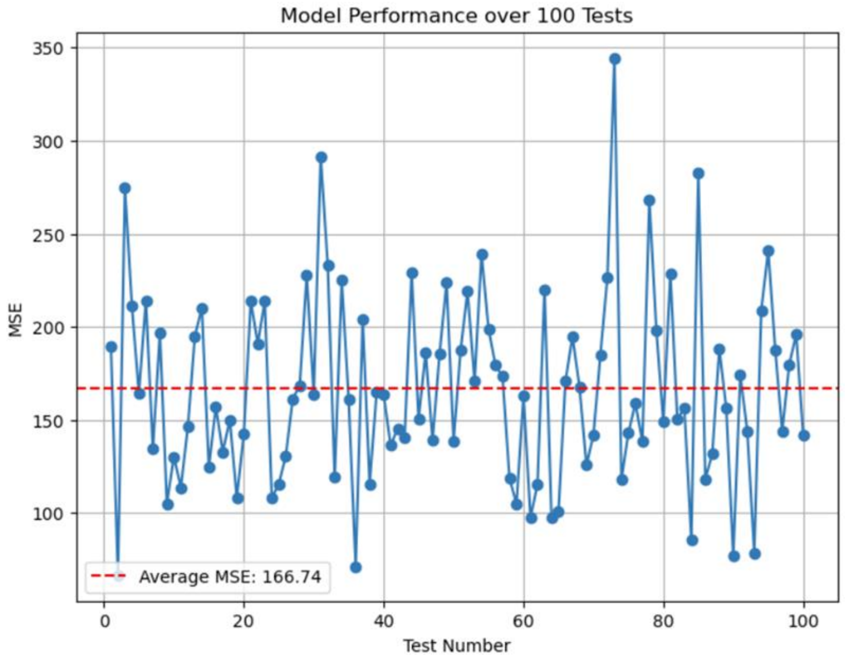
max_features



Random Forest

Model:
model = GridSearchCV(estimator=RandomForestRegressor(), param_grid=param_grid, cv=5, scoring='neg_mean_squared_error')
model.fit(X, y)

Best Model:
max_depth = 20, max_features = 'sqrt', min_samples_leaf = 1

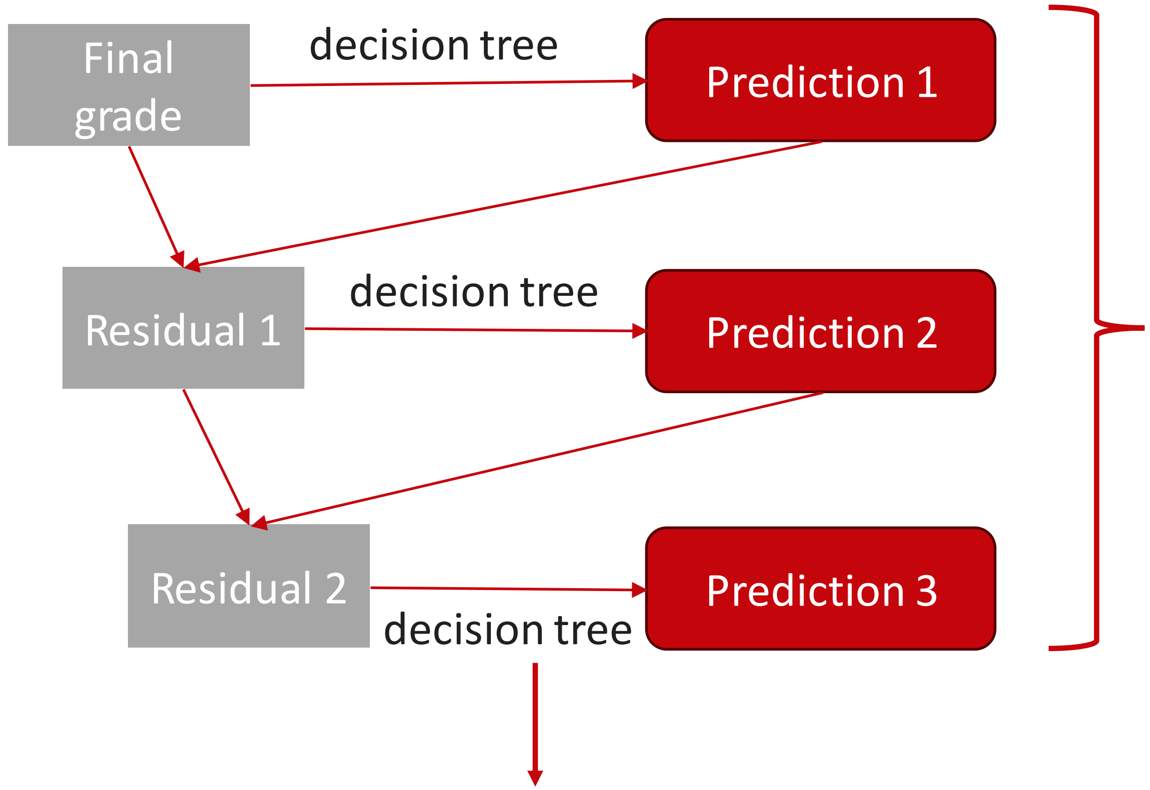


Gradient Boosting

Gradient Boosting iteratively training weak predictive models, focusing on the residuals of the previous round.

Key parameters：

learning_rate



n_estimators

max_depth

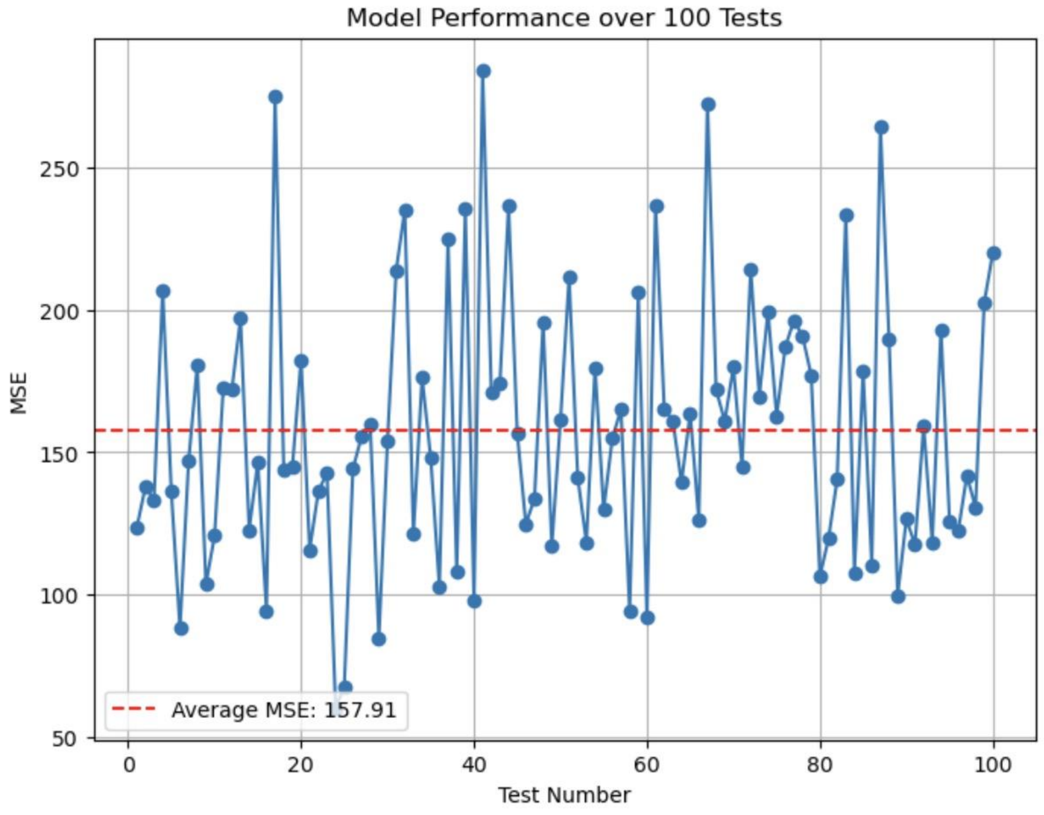
Gradient Boosting

Model:

model = GridSearchCV(estimator=GradientBoostingRegressor(), param_grid=param_grid, cv=5, scoring='neg_mean_squared_error')
model.fit(X, y)

Best Model:

learning_rate = 0.1,
max_depth = 1,
n_estimators = 100



Model improvements

Normalizing the data and removing columns with negative lasso coefficients.

The MSE values：

Random Forest：

	standardize	Non-Standardize
Remove negative coefficients of lasso	148.32	172.17
reserve negative coefficients of lasso	155.88	155.24

Gradient Descent：

	standardize	Non-Standardize
Remove negative coefficients of lasso	151.44	155.67
reserve negative coefficients of lasso	156.11	170.65

We can see that both ways can improve the model.

(In this project, we've used the normalized data.)

Conclusion and Model Application

Models:

RandomForestRegressor(max_depth = 20, max_features = 'sqrt', min_samples_leaf = 1)

GradientBoostingRegressor(learning_rate = 0.1, max_depth = 1, n_estimators = 100)

Predict final grade for this term:

	Q 3	Q 4	Q 6	Q 7	HW 1	HW 2	HW 3	HW 4	HW 5	Pre	Pre Feedback	Pre Feedback 2	Midterm Exam	FinalGrade (by Random Forest)	Final Grade (by Grediant Boosting)
Student 1	12	16	10	3	20	20	20	20	20	24.5	2	2	80	60	55
Student 2	12	16	10	3	20	19	20	20	20	24	2	2	75	48	45
student 3	12	16	10	3	20	20	19	20	18	25	2	2	66	44	38

*The predicted score is a standardized z-score out of 75.

*The MSE of the model is high, so the prediction is not very accurate.

Predictions of our final grades reflect our past academic performance, but they don't define our future. With determination and effort, we have the power to shape and improve our outcomes in the upcoming final exams.

Wishing everyone good results in the final exams!

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON