Data Science I: Homework 2

Neal Kuperman

February 3, 2026

Problem 1: College Data Analysis

Consider the `College` data from the ISLP package. Details about the data is described on page 65 of the ISLP textbook for this class (https://islp.readthedocs.io/en/latest/datasets/College.html). We would like to ***predict*** the number of applications received using the other variables. 80% of the data (randomly generated) will be treated as training data. The rest will be the test data.

(a) Fit a linear model using least squares and report the estimate of the test error.

(b) Fit a tree to the data. Summarize the results. Unless the number of terminal nodes is large, display the tree graphically. Report its MSE.

(c) Use Cross validation to determine whether pruning is helpful and determine the optimal size for the pruned tree. Compare the pruned and un-pruned trees. Report MSE for the pruned tree. Which predictors seem to be the most important?

(d) Use a bagging approach to analyze the data with $B = 500$ and $B = 1000$. Compute the MSE. Which predictors seem to be the most important?

(e) Repeat (d) with a random forest approach with $B = 500$ and $B = 1000$, and $m \approx p = 3$.

(f) Compare the results from the various methods. Which method would you recommend?

Note(s)

The data is statistics for a large number of US Colleges from the 1995 issue of US News and World Report. Table 1 contains the variable descriptions, which can also be found on the ISLP documentation. Tables 2 - 4 contain the descriptive statistics for the data.

Variable	Description
Private	A factor with levels No and Yes indicating private or public university
Apps	Number of applications received
Accept	Number of applications accepted
Enroll	Number of new students enrolled
Top10perc	Pct. new students from top 10% of H.S. class
Top25perc	Pct. new students from top 25% of H.S. class
F.Undergrad	Number of full time undergraduates
P.Undergrad	Number of part time undergraduates
Outstate	Out-of-state tuition
Room.Board	Room and board costs
Books	Estimated book costs
Personal	Estimated personal spending
PhD	Pct. of faculty with Ph.D.s
Terminal	Pct. of faculty with terminal degree
S.F.Ratio	Student/faculty ratio
perc.alumni	Pct. alumni who donate
Expend	Instructional expenditure per student
Grad.Rate	Graduation rate

Table 1: College Dataset Variables

	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad
count	777.00	777.00	777.00	777.00	777.00	777.00
mean	3001.64	2018.80	779.97	27.56	55.80	3699.91
std	3870.20	2451.11	929.18	17.64	19.80	4850.42
min	81.00	72.00	35.00	1.00	9.00	139.00
255075max	48094.00	26330.00	6392.00	96.00	100.00	31643.00

Table 2: College Dataset Descriptive Statistics (Part 1)

	P.Undergrad	Outstate	Room.Board	Books	Personal	PhD
count	777.00	777.00	777.00	777.00	777.00	777.00
mean	855.30	10440.67	4357.53	549.38	1340.64	72.66
std	1522.43	4023.02	1096.70	165.11	677.07	16.33
min	1.00	2340.00	1780.00	96.00	250.00	8.00
255075max	21836.00	21700.00	8124.00	2340.00	6800.00	103.00

Table 3: College Dataset Descriptive Statistics (Part 2)

	Terminal	S.F.Ratio	perc.alumni	Expend	Grad.Rate
count	777.00	777.00	777.00	777.00	777.00
mean	79.70	14.09	22.74	9660.17	65.46
std	14.72	3.96	12.39	5221.77	17.18
min	24.00	2.50	0.00	3186.00	10.00
255075max	100.00	39.80	64.00	56233.00	118.00

Table 4: College Dataset Descriptive Statistics (Part 3)

Solution

Due to the wide range of values in the college data, the college data was scaled using the standard scalar transform from scikit-learn.

1 a)

Dep. Variable:	y	**R-squared:**	0.923
Model:	OLS	**Adj. R-squared:**	0.921
Method:	Least Squares	**F-statistic:**	370.2
Date:	Mon, 02 Feb 2026	**Prob (F-statistic):**	3.18e-279
Time:	18:37:21	**Log-Likelihood:**	-4582.2
No. Observations:	543	**AIC:**	9200.
Df Residuals:	525	**BIC:**	9278.
Df Model:	17		
Covariance Type:	nonrobust		

	coef	std err	t	P> |t|	[0.025	0.975]
Intercept	3367.3648	144.649	23.280	0.000	3083.204	3651.526
Accept	3902.1993	124.078	31.449	0.000	3658.448	4145.951
Enroll	-726.6304	225.644	-3.220	0.001	-1169.906	-283.355
Top10perc	1014.8151	133.744	7.588	0.000	752.077	1277.553
Top25perc	-361.2881	118.756	-3.042	0.002	-594.584	-127.993
F.Undergrad	310.4624	203.903	1.523	0.128	-90.104	711.029
P.Undergrad	45.9057	68.948	0.666	0.506	-89.541	181.353
Outstate	-366.4795	100.632	-3.642	0.000	-564.171	-168.788
Room.Board	185.0855	70.459	2.627	0.009	46.670	323.501
Books	-20.3741	51.773	-0.394	0.694	-122.081	81.333
Personal	9.8714	55.221	0.179	0.858	-98.610	118.353
PhD	-160.0626	95.706	-1.672	0.095	-348.077	27.951
Terminal	-28.3167	94.736	-0.299	0.765	-214.426	157.792
S.F.Ratio	57.4472	65.785	0.873	0.383	-71.787	186.682
perc.alumni	-1.0088	65.600	-0.015	0.988	-129.880	127.862
Expend	440.0177	82.356	5.343	0.000	278.231	601.805
Grad.Rate	161.8552	66.096	2.449	0.015	32.009	291.701
Private_Yes	-380.2813	187.179	-2.032	0.043	-747.993	-12.570

The test MSE for the linear regression model is $642,753.9$ and the test R^2 is 0.946.

1 b)

Two regression tree models with maximum depths of 3 and 10 were fit to the data to establish baseline performance. Training and test MSE and R^2 for the two trees are given in table 5. Figure 1 shows the tree with max depth of 3. The tree with a max depth of 10, not displayed due to its complexity, outperforms the smaller tree in both test MSE and test R^2. The low training MSE and high training R^2 suggest that the tree with a max depth of 10 is overfitting the training data, however, it is still able to generalize well to the test data. The most important feature for both trees is the numbers of applications accepted and the percentage of new students from the top 10% of the high school class (Table 6).

Max Depth	**Training MSE**	**Test MSE**	**Training** R^2	**Test** R^2
3	1.579×10^6	1.952×10^6	0.90	0.84
10	$9,582$	1.0699×10^6	0.999	0.91

Table 5: MSE and R^2 for Regression Trees with Different Max Depths

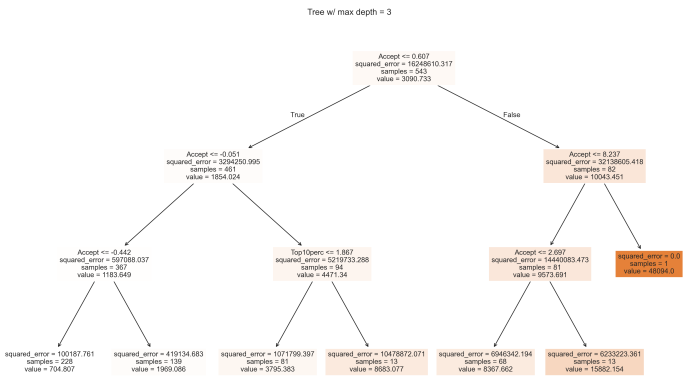


Figure 1: Tree with Max Depth of 3

Max Depth = 3	
Feature	**Importance**
Accept	0.966403
Top10perc	0.033597
Intercept	0.000000
Personal	0.000000
Grad.Rate	0.000000
Expend	0.000000
perc.alumni	0.000000
S.F.Ratio	0.000000
Terminal	0.000000
PhD	0.000000

Max Depth = 10	
Feature	**Importance**
Accept	0.910615
Top10perc	0.038446
Top25perc	0.016548
F.Undergrad	0.015113
Outstate	0.003668
Expend	0.002848
PhD	0.002825
Grad.Rate	0.002648
S.F.Ratio	0.002398
Books	0.001412

Table 6: Feature Importance for Regression Trees

1 c)

To evaluate the effect of pruning, we applied cost-complexity pruning to the tree with maximum depth 10. Figure 2 shows the pruned best tree, which has an alpha value of 17,157.520. Table 7 shows the training and test MSE and R^2 for both the unpruned and pruned trees. Pruning improved test MSE and test R^2, indicating reduced overfitting. Table 8 shows the feature importance for the pruned tree. The top six features (Accept, Top10perc, Top25perc, F.Undergrad, Outstate, and Expend) are identical to those in the unpruned tree. Personal is the only additional feature with non-zero importance in the pruned tree that was not among the top features in the unpruned model.

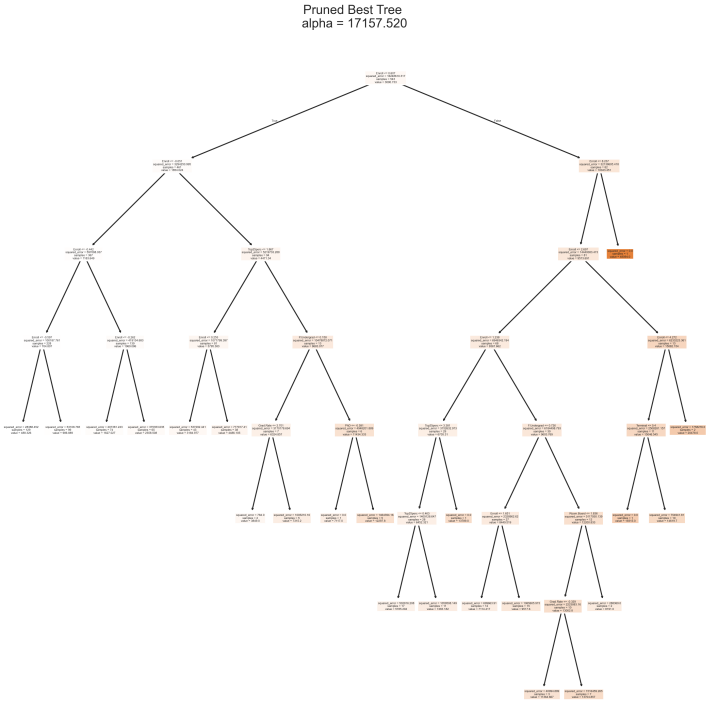


Figure 2: Pruned Best Tree

Tree Type	**Training MSE**	**Test MSE**	**Training** R^2	**Test** R^2
Unpruned	$9,582$	1.0699×10^6	0.999	0.91
Pruned	$332,673$	$642,753$	0.980	0.93

Table 7: MSE and R^2 for Regression Trees with Different Max Depths

Feature	Importance
Accept	0.924818
Top10perc	0.039207
Top25perc	0.013527
F.Undergrad	0.009759
Outstate	0.004465
Expend	0.003317
Personal	0.002588
PhD	0.002319
Intercept	0.000000
S.F.Ratio	0.000000
Grad.Rate	0.000000
perc.alumni	0.000000
Books	0.000000
Terminal	0.000000
Room.Board	0.000000
P.Undergrad	0.000000
Enroll	0.000000
Private_Yes	0.000000

Table 8: Feature Importance for Pruned Regression Tree

1 d)

Table 9 shows the training and test MSE and R^2 for the regression trees with bagging for B values of 500 and 1000. Both trees have similar training and test MSE and R^2, indicating that a value of b=500 is sufficient for this data. Table 10 shows the feature importance for the regression trees with bagging for B values of 500 and 1000. The top six features (Accept, Enroll, Top10perc, Top25perc, Expend, and Grad.Rate) are the same for both trees.

B value	**Training MSE**	**Test MSE**	**Training** R^2	**Test** R^2
500	351,729.95	655,166.13	0.978	0.945
1000	325,604.62	652,137.78	0.980	0.945

Table 9: MSE and R^2 for Regression Trees with Bagging for Different B Values

Bagging (B=500)	
Feature	**Importance**
Accept	0.802421
Enroll	0.103269
Top10perc	0.023217
Top25perc	0.017734
Expend	0.009980
Grad.Rate	0.008378
F.Undergrad	0.005095
S.F.Ratio	0.004847
perc.alumni	0.004394
Outstate	0.004204

Bagging (B=1000)	
Feature	**Importance**
Accept	0.798977
Enroll	0.108115
Top10perc	0.022769
Top25perc	0.018266
Expend	0.008796
Grad.Rate	0.008535
F.Undergrad	0.005027
S.F.Ratio	0.004975
perc.alumni	0.004345
Outstate	0.004221

Table 10: Feature Importance for Regression Trees with Bagging (B=500 and B=1000)

1 e)

Number of Estimators	**Training MSE**	**Test MSE**	**Training** R^2	**Test** R^2
500	$418,695.37$	$1,055,332.13$	0.974	0.911
1000	$386,329.45$	$1,086,990.94$	0.976	0.909

Table 11: MSE and R^2 for Regression Trees with Random Forest for Different Number of Estimators

Bagging (B=500)	
Feature	**Importance**
Accept	0.259394
Enroll	0.194531
F.Undergrad	0.152924
P.Undergrad	0.055010
Top25perc	0.043281
PhD	0.038476
Top10perc	0.036371
Private_Yes	0.031446
Expend	0.029643
Terminal	0.027510

Bagging (B=1000)	
Feature	**Importance**
Accept	0.252276
Enroll	0.193074
F.Undergrad	0.161521
P.Undergrad	0.050646
Top25perc	0.045768
Top10perc	0.039110
PhD	0.035762
Private_Yes	0.031904
Expend	0.029108
Terminal	0.028647

Table 12: Feature Importance for Regression Trees with Random Forest (n estimators=500 and n estimators=1000)

1 f)

I would recommend the random forest with bagging classifier. Although the linear regression model has a similar MSE for the test data, the random forest will work with non-linear relationships making it a more robust classifier. The major benefit to the linear regression model is the interpretability of the coefficients.

Method	**Training MSE**	**Test MSE**	**Training** R^2	**Test** R^2
Linear Regression	$1,251,247.29$	$642,753.89$	0.923	0.946
Best Regression Tree (alpha=17,157.520)	$9,582$	$1,069,900$	0.999	0.911
Pruned Best Regression Tree	$332,673$	$642,753$	0.980	0.93
Bagging, B=500	$351,729.95$	$655,166.13$	0.978	0.945
Bagging, B=1000	$325,604.62$	$652,137.78$	0.980	0.945
Random Forest, n estimators=500	$418,695.37$	$1,055,332.13$	0.974	0.911
Random Forest, n estimators=1000	$386,329.45$	$1,086,990.94$	0.976	0.909

Table 13: MSE and R^2 for Different Regression Methods

Problem 2: Admission Data Analysis

Consider the business school admission data available in the admission.csv. The admission officer of a business school has used an "*index*" of undergraduate grade point average (GPA,X1) and graduate management aptitude test (GMAT,X2) scores to help decide which applicants should be admitted to the school's graduate programs. This index is used to categorize each applicant into one of three groups - admit (group 1), do not admit (group 2), and borderline (group 3). We will take the last ***four*** observations in ***each category*** as test data and the remaining observations as training data.

(a) Perform an exploratory analysis of the training data by examining appropriate plots and comment on how helpful these predictors may be in predicting response.

(b) Perform an LDA using the training data. Superimpose the decision boundary on an appropriate display of the data. Does the decision boundary seem sensible? In addition, compute the confusion matrix and overall misclassification rate based on both training and test data. What do you observe?

(c) Repeat (b) using QDA.

(d) Fit a KNN with K chosen optimally using test error rate. Report error rate, sensitivity, specificity, and AUC for the optimal KNN based on the training data. Also, report its estimated test error rate.

(e) Compare the results in (b), (c) and (d). Which classifier would you recommend? Justify your conclusions.

Note(s)

Test/train data was split using the following code:

```
train_data = admin_data[admin_data.groupby('Group').cumcount(
    ascending=False) >= 4]
test_data = admin_data.groupby('Group').tail(4)
```

The following metrics are useful for evaluating the performance of classifiers. We will use some of them when evaluating the performance of the KNN classifier.

Error rate: percentage misclassification

$$\frac{\text{Misclassified}}{\text{Total}} = 1 - \text{Accuracy}$$

Sensitivity (*true positive rate*): measures the proportion of actual positives correctly identified as positive

$$\frac{\text{Number of true positives}}{\text{Number of true positives} + \text{Number of false negatives}}$$

Specificity (*true negative rate*): measures the proportion of actual negatives correctly identified. Note: Also called recall.

$$\frac{\text{Number of true negatives}}{\text{Number of true negatives} + \text{Number of false positives}}$$

Precision: Ratio of the correctly predicted class to the total predicted class

$$\frac{\text{Number of true positives}}{\text{Number of true positives} + \text{Number of false positives}}$$

F1 Score: A harmonic mean between the Precision and Recall score

$$2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

ROC Curve: A Receiver Operating Characteristic (ROC) curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system by plotting the true-positive rate (sensitivity) against the false-positive rate ($1 -$ specificity) across various decision thresholds [**?**].

Solution

2 a)

The following table and plots were generated to explore the admission training data

- Table 14: Descriptive statistics for the admission data.
- Figure 3: Distribution plots for the GPA, GMAT, and Group.
- Figure 4: Scatter plot of GPA vs GMAT.
- Figure 5: Box plots for the GPA and GMAT by Group.

We can see that the data is relatively balanced across groups and that there is good separation between the groups (see Figure 4). Additionally, There is not a large variation in the distribution of GPA and GMAT within and across each group. In particular, the scatter plot shows that, given a good classifier, we should be able to separate the groups well.

	GPA	GMAT	Group
count	85	85	85
mean	2.97	488.45	1.94
std	0.42	81.52	0.82
min	2.13	313	1
25%	2.60	425	1
50%	3.01	482	2
75%	3.30	538	3
max	3.80	693	3

Table 14: Admission Data Descriptive Statistics

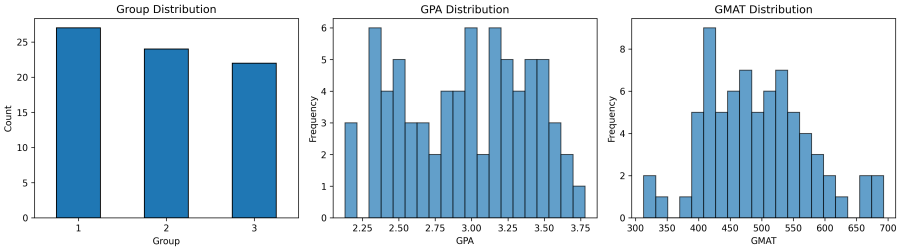


Figure 3: Training Admission Data - Distribution Plots

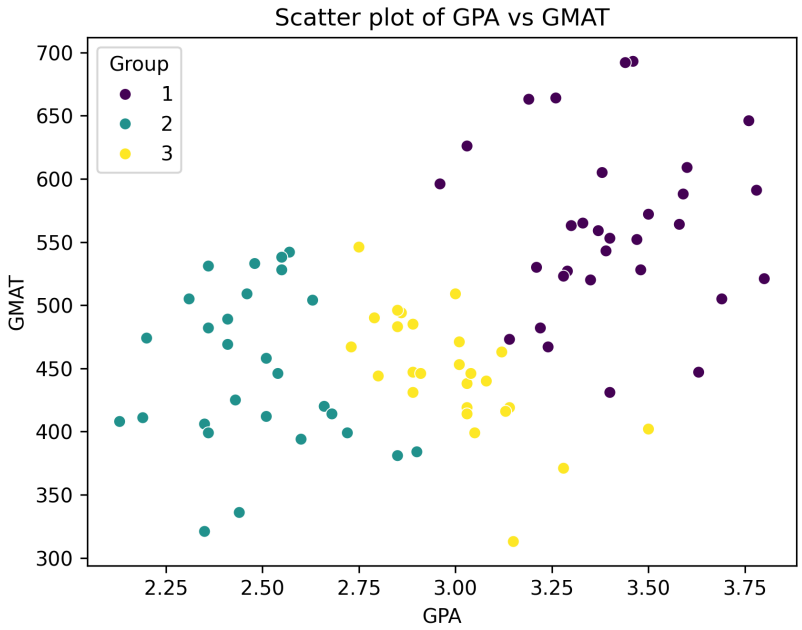


Figure 4: Training Admission Data - Scatter Plot of GPA vs GMAT

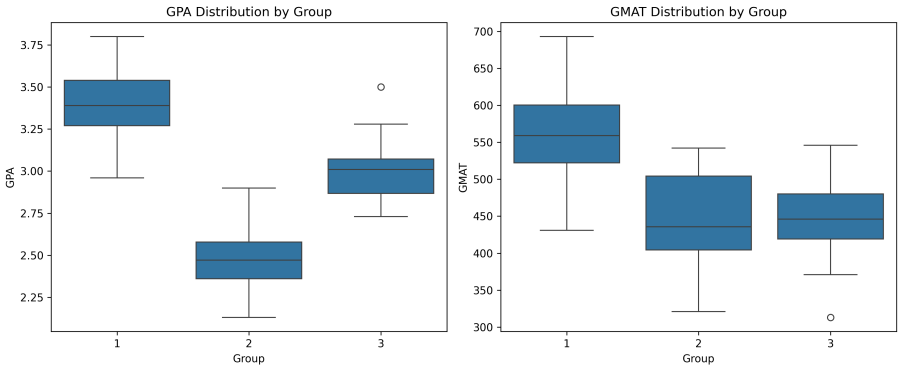


Figure 5: Training Admission Data - Box Plots of GPA and GMAT by Group

2 b)

Figure 6 shows the Admission Data with LDA decision boundaries. The boundaries are look reasonable, but there is room for improvement as the misclassification rate for the test data was 25%. From the test data, we can see that the LDA has the most trouble classifying group 2, however three of the four test data points live on or near the decision boundary. It may be worthwhile exploring different train/test splits to study how much role the sampling has on the classifier's performance. Tables 15, 16, and 17 show the confusion matrices for the training and test data respectively and the misclassification rates for the training and test data.

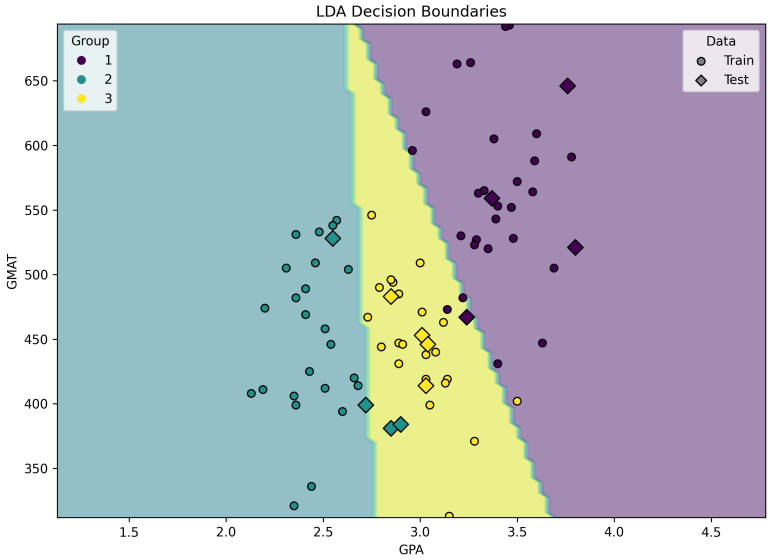


Figure 6: Admission Data with LDA Decision Boundaries

Truth Predicted	1	2	3
1	25	0	1
2	0	24	0
3	2	0	21

Table 15: LDA Confusion Matrix for Training Data

Truth Predicted	1	2	3
1	3	0	0
2	0	2	0
3	1	2	4

Table 16: LDA Confusion Matrix for Test Data

Data	Misclassification Rate
Test	25.00%
Train	4.11%
Total	7.06%

Table 17: LDA Misclassification Rates

2 c)

Figure 7 shows the Admission Data with QDA decision boundaries. The quadratic boundaries provide better separation between classes compared to the linear boundaries from LDA. This is further reflected in the confusion matrices (tables 18 and 19) and misclassification rates (table 20). Going from LDA to QDA, the misclassification rate for the test data decreased from 25% to 16.67% and the total misclassification rate decreased from 7.06% to 4.71%.

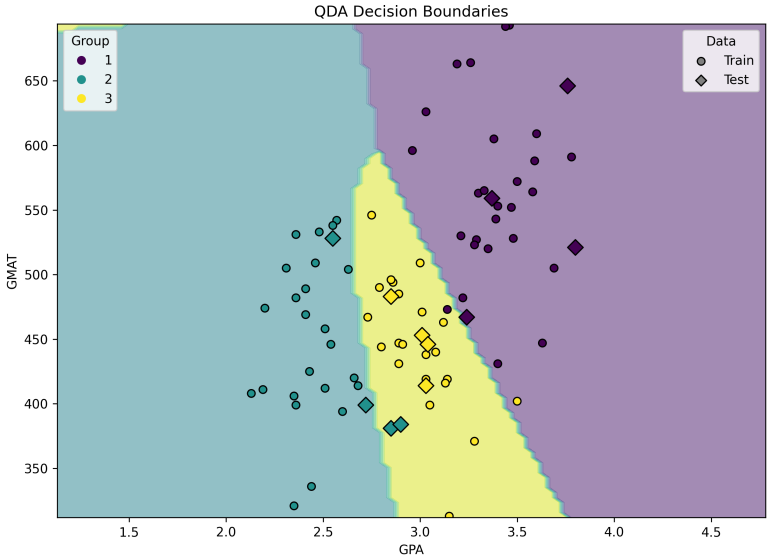


Figure 7: Admission Data with LDA Decision Boundaries

Truth Predicted	1	2	3
1	26	0	1
2	0	24	0
3	1	0	21

Table 18: QDA Confusion Matrix for Training Data

Truth Predicted	1	2	3
1	4	0	0
2	0	2	0
3	0	2	4

Table 19: QDA Confusion Matrix for Test Data

Data	**Misclassification Rate**
Test	16.67%
Train	2.74%
Total	4.71%

Table 20: QDA Misclassification Rates

2 d)

For a KNN classifier, the predicted probability for a class is the proportion of the K neighbors belonging to that class, i.e.,

$$P(\text{class } k \mid x) = \frac{\text{number of neighbors in class } k}{K}$$

The ROC curve is generated by varying the probability threshold required to classify a point as positive. When building the ROC curve, we vary $t \in [0, 1]$ with the decision rule: predict Class n if $P(\text{Class } n \mid x) \geq t$.

For a multiclass classification problem, we can no longer generate a single ROC curve since it is based on a binary classifier. We can use the one-vs-all scheme, which compares each class against all the others (combined as one)[1].

NOTE: Since KNN is a distance based method, we want to make sure that GMAT and GPA are on similar scales. We are going to use the standard scaling method from scikit learn on our training and test data. All plots are generated using data transformed back into the original coordinates.

[1] https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html

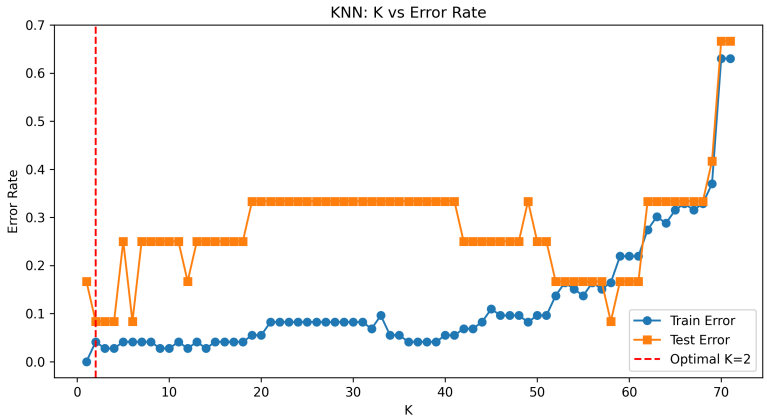


Figure 8: KNN: K vs Error Rate

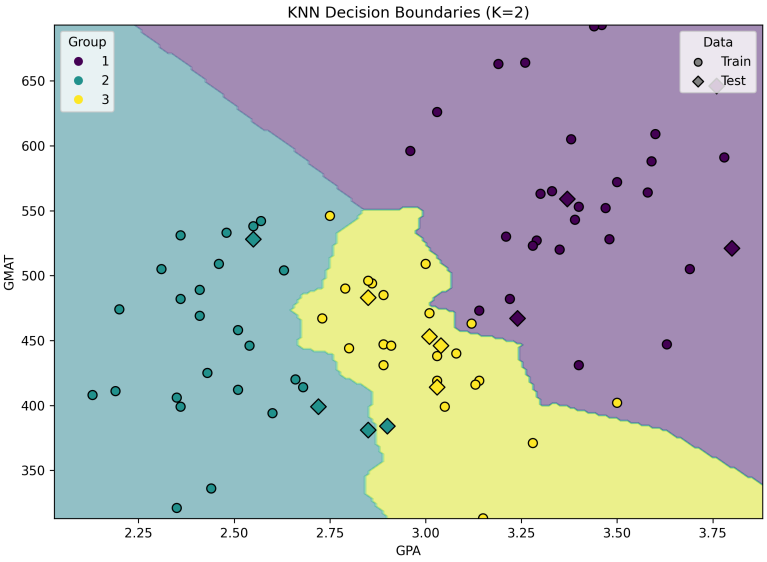


Figure 9: KNN: Decision Boundaries

Truth Predicted	1	2	3
1	27	0	0
2	0	24	0
3	2	1	19

Table 21: KNN Confusion Matrix for Training Data, optimal K=2

Truth Predicted	1	2	3
1	4	0	0
2	0	3	1
3	0	0	4

Table 22: KNN Confusion Matrix for Test Data, optimal K=2

Class	**Sensitivity**	**Specificity**
1	1.000	0.9565
2	1.000	0.9796
3	0.8636	1.0000

Metric	**Value**
AUC (Macro OvR)	0.9986
Test Accuracy	0.9167
Test Error Rate	0.0833

Table 23: KNN Classification Metrics

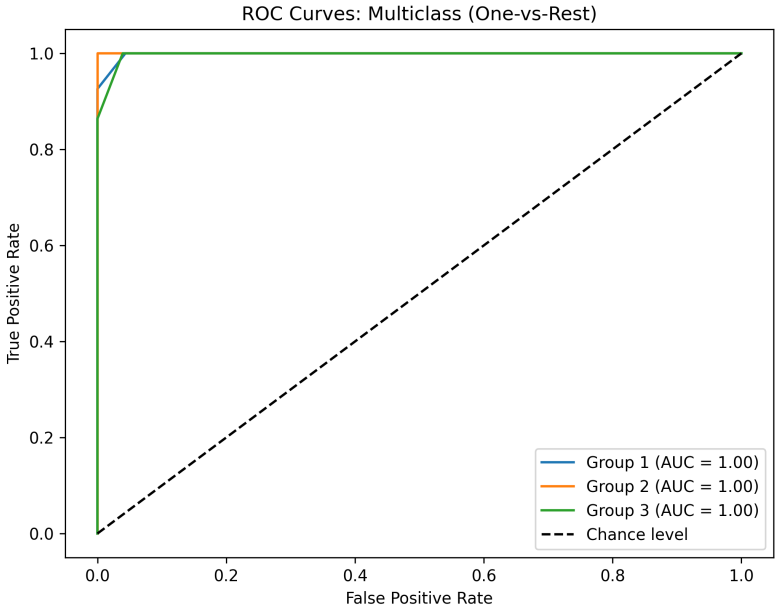
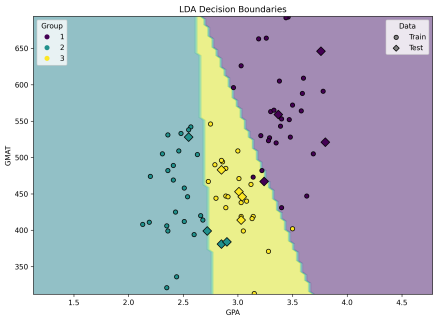


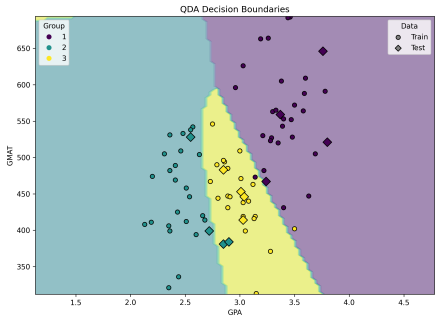
Figure 10: KNN: ROC Curve

2 e)

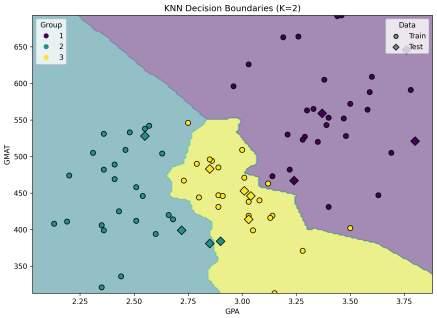
Fig 11 shows the decision boundaries for LDA, QDA, and KNN and tables 24 and 25 show misclassification rates and confusion matrices for the training and test data respectively. We can see that the KNN classifier outperforms the other two classifiers, with the KNN having a misclassification rate for the test data of 8.3% compared to 16.7% and 25% for QDA and LDA respectively. Given the lower test error rate and overall error rate, I would recommend the KNN classifier.



(a) LDA



(b) QDA



(c) KNN

Figure 11: Decision Boundaries for LDA, QDA, and KNN

Method	**Train Error**	**Test Error**	**Total Error**
LDA	4.11%	25.00%	7.06%
QDA	2.74%	16.67%	4.71%
KNN (K=2)	4.11%	8.3%	4.7%

Table 24: Misclassification Rates by Classification Method

	Training			**Test**		
	Truth			Truth		
	1	2	3	1	2	3
LDA						
Predicted						
1	25	0	1	3	0	0
2	0	24	0	0	2	0
3	2	0	21	1	2	4
QDA						
Predicted						
1	26	0	1	4	0	0
2	0	24	0	0	2	0
3	1	0	21	0	2	4
KNN (K=2)						
Predicted						
1	27	0	0	4	0	0
2	0	24	0	0	3	1
3	2	1	19	0	0	4

Table 25: Confusion Matrices by Classification Method