MLDS CENTER
Maryland Longitudinal Data System
Better Data • Informed Choices • Improved Results

MLDS Center Research Series

Applications of Data Science Methods to MLDS Data

Brennan Register, Patrick Sheehan,
& Tracy Sweet

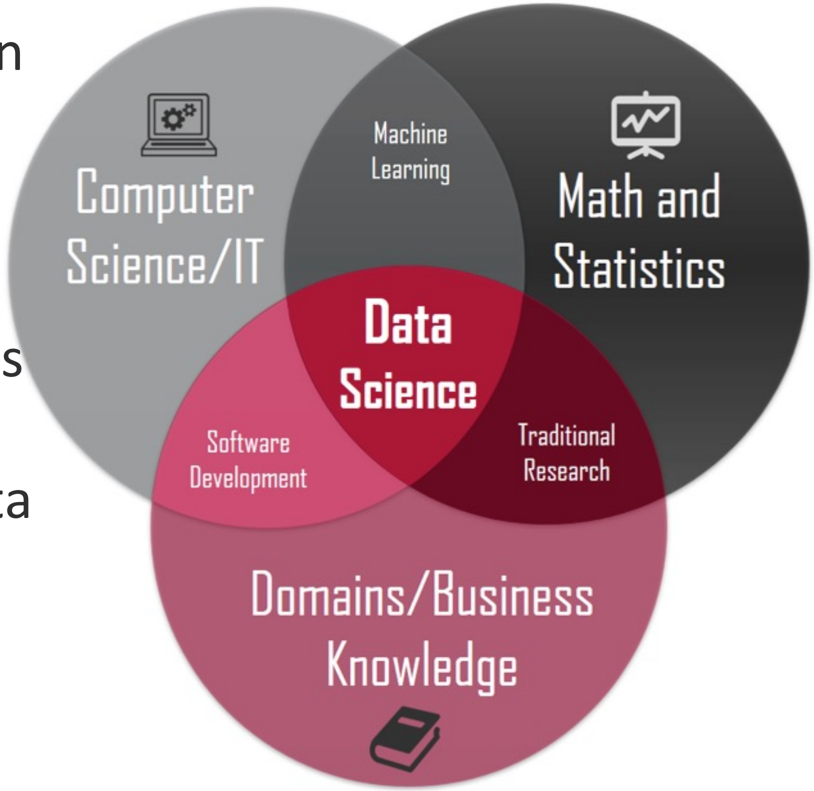
May 5, 2022

Outline

- Introduction to Common Data Science Methods
- Example with Simulated Data
- MLDS Application

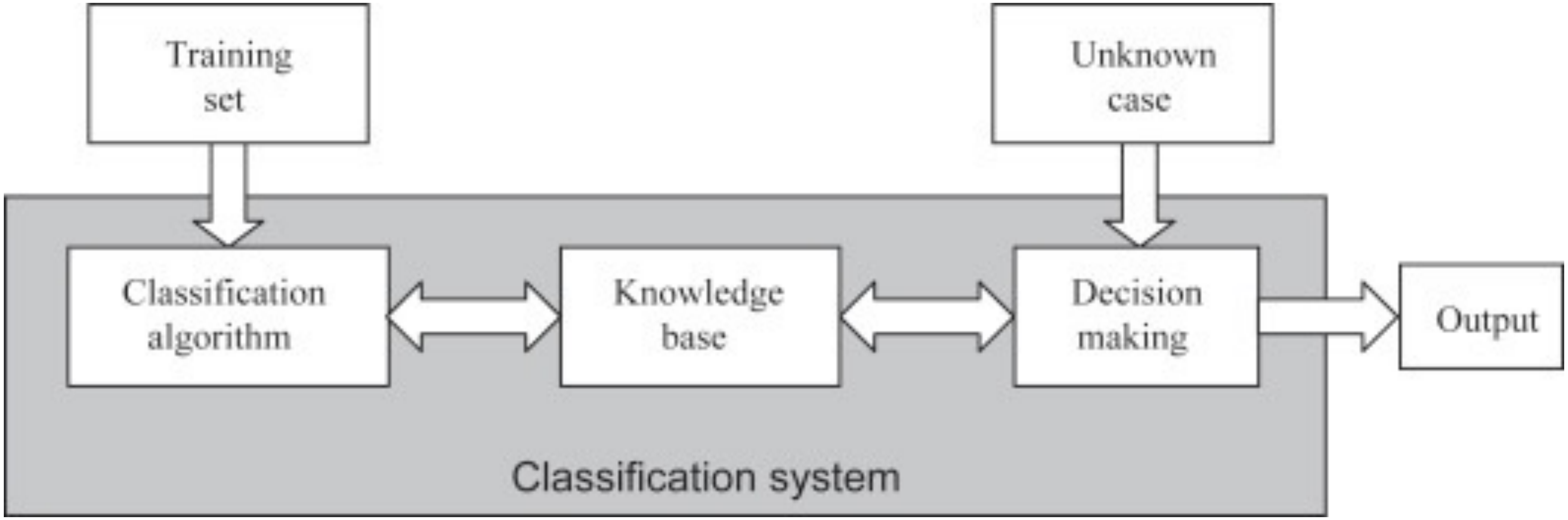
Data Science & Machine Learning

- Data science is the intersection of computer science, statistics and a content area
- Machine Learning (ML) focuses on building computer algorithms that learn from data
- The algorithms are fine-tuned and then applied to data



MLDS Center
Maryland Longitudinal Data System

General Idea



Common ML Output Types

Regression

Predict numerical values
(e.g. price of house)

Classification

One of n labels…
(cat, dog, human)

Clustering

Most similar other examples
(e.g. related products on Amazon)

Sequence Prediction

What comes next?
“If you want something done _____, do it yourself”

Two Main Approaches

Supervised Learning

- Labeled datasets
  - Outcome **Y**
- p predictors **X**
- When Y is quantitative → **regression problem**
- When Y is categorical → **classification problem**

Unsupervised Learning

- Unlabeled datasets
  - No outcome variable
- Discover hidden patterns in data
- Three main tasks: clustering, association and dimensionality reduction

Simulated Data Example

- Predicting graduate school admissions given a set of student characteristics
- Sample of 500 students
- Classification problem
- Supervised Learning

Variables in Simulated Data

- Admitted to Grad School (either 0 or 1)

Outcome

- GRE Scores (out of 340)
- TOEFL Scores (out of 120)
- University Rating (out of 5)
- Statement of Purpose (out of 5)
- Letter of Recommendation Strength (out of 5)
- Undergraduate College GPA (out of 4)
- Research Experience (either 0 or 1)
- Male (either 0 or 1)

Predictors

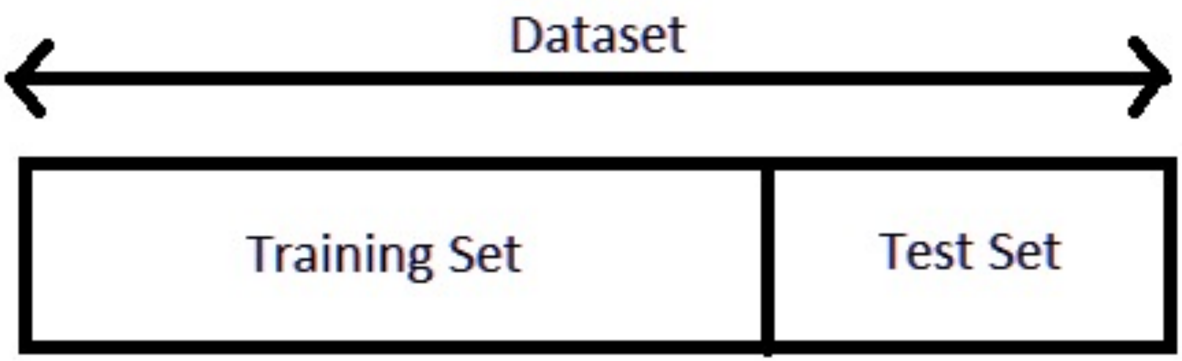
Snapshot of the Simulated Dataset

	GRE.Score	TOEFL.Score	University.Rating	SOP	LOR	CGPA	Research	Male	Admit
1	337	118	4	4.5	4.5	3.73	1	1	1
2	324	107	4	4.0	4.5	2.95	1	0	1
3	316	104	3	3.0	3.5	2.08	1	0	0
4	322	110	3	3.5	2.5	2.75	1	0	1
5	314	103	2	2.0	3.0	2.29	0	1	0
6	330	115	5	4.5	3.0	3.42	1	1	1

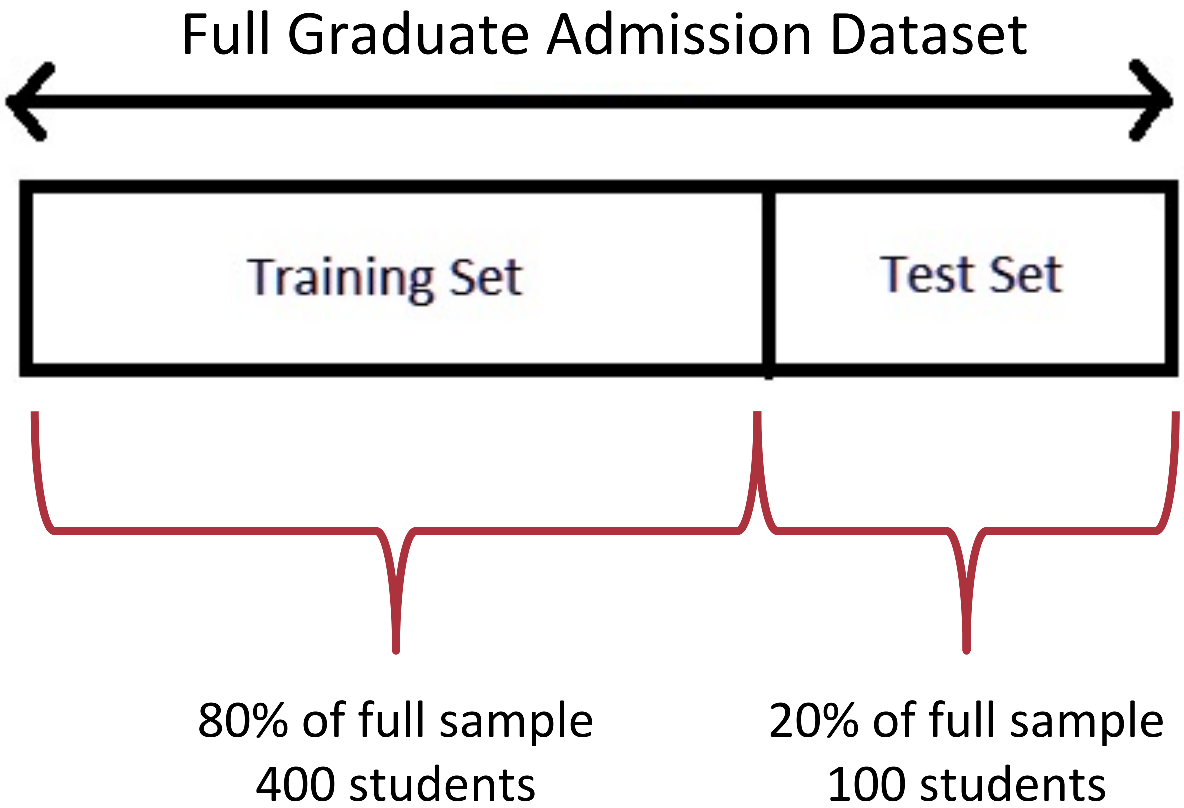
*note this is not real data

Training vs Testing

- **Training Set:** *The sample of data used to fit the model*
- **Testing Set:** *The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset*

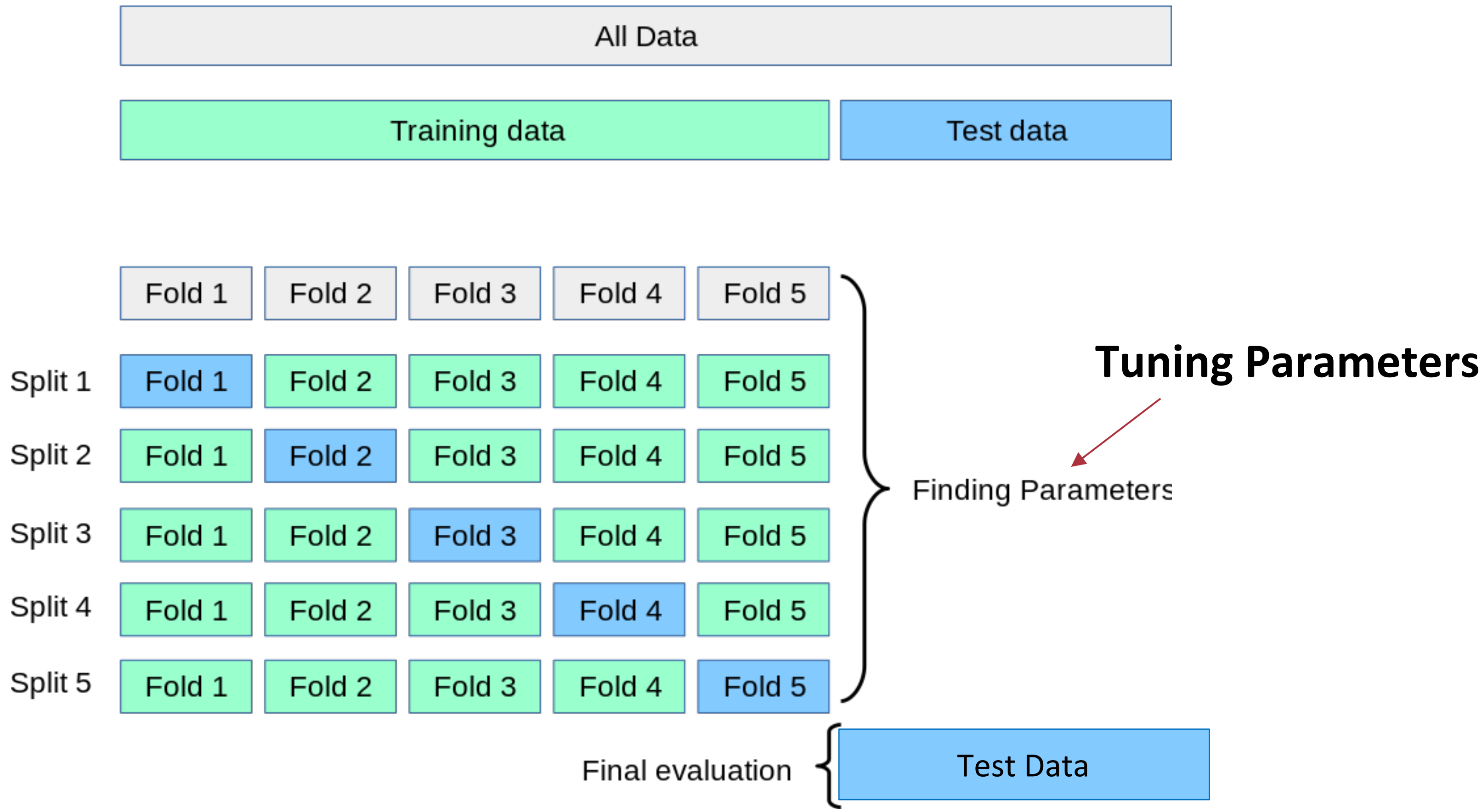


Training and Testing Sets for Grad Admission

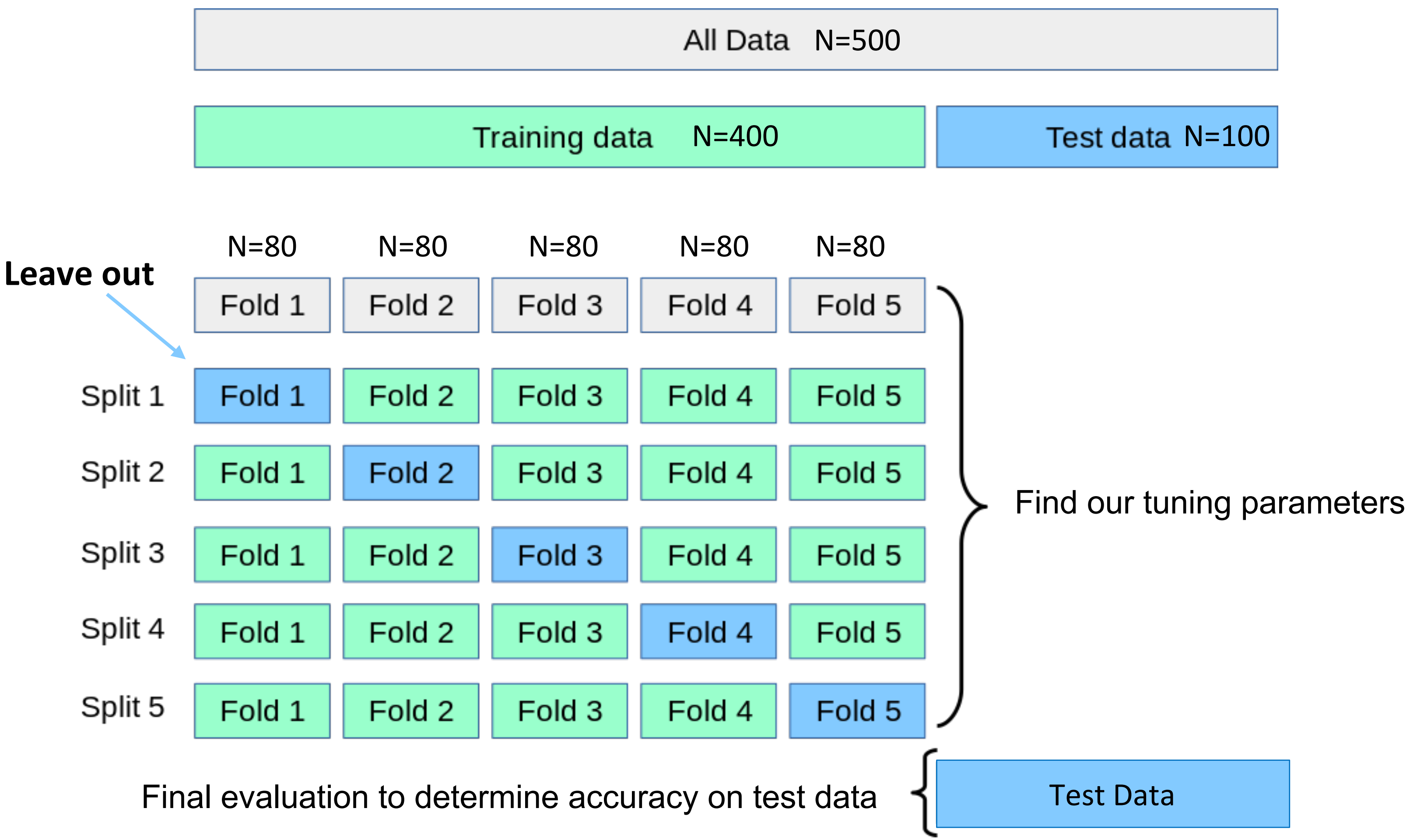


MLDS Center
Maryland Longitudinal Data System

K-fold Cross Validation



5-fold Cross Validation for Grad Admission



MLDS Center
Maryland Longitudinal Data System

Model Evaluation

- **Accuracy:** a measure of how well the machine learning model performs

- Continuous Y: Mean Squared Error
- Categorical Y: Misclassification Rate

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

		Predicted	
		Good	Bad
Actual	Good	True Positive (d)	False Negative (c)
	Bad	False Positive (b)	True Negative (a)

Example Confusion Matrix for Grad Admission

- Random Forest

Confusion Matrix		Truth	
		Not Admitted	Admitted
Prediction	Not Admitted	55	10
	Admitted	5	30

Accuracy:

(55 + 30) / 100 = 85%

Bias - Variance Tradeoff

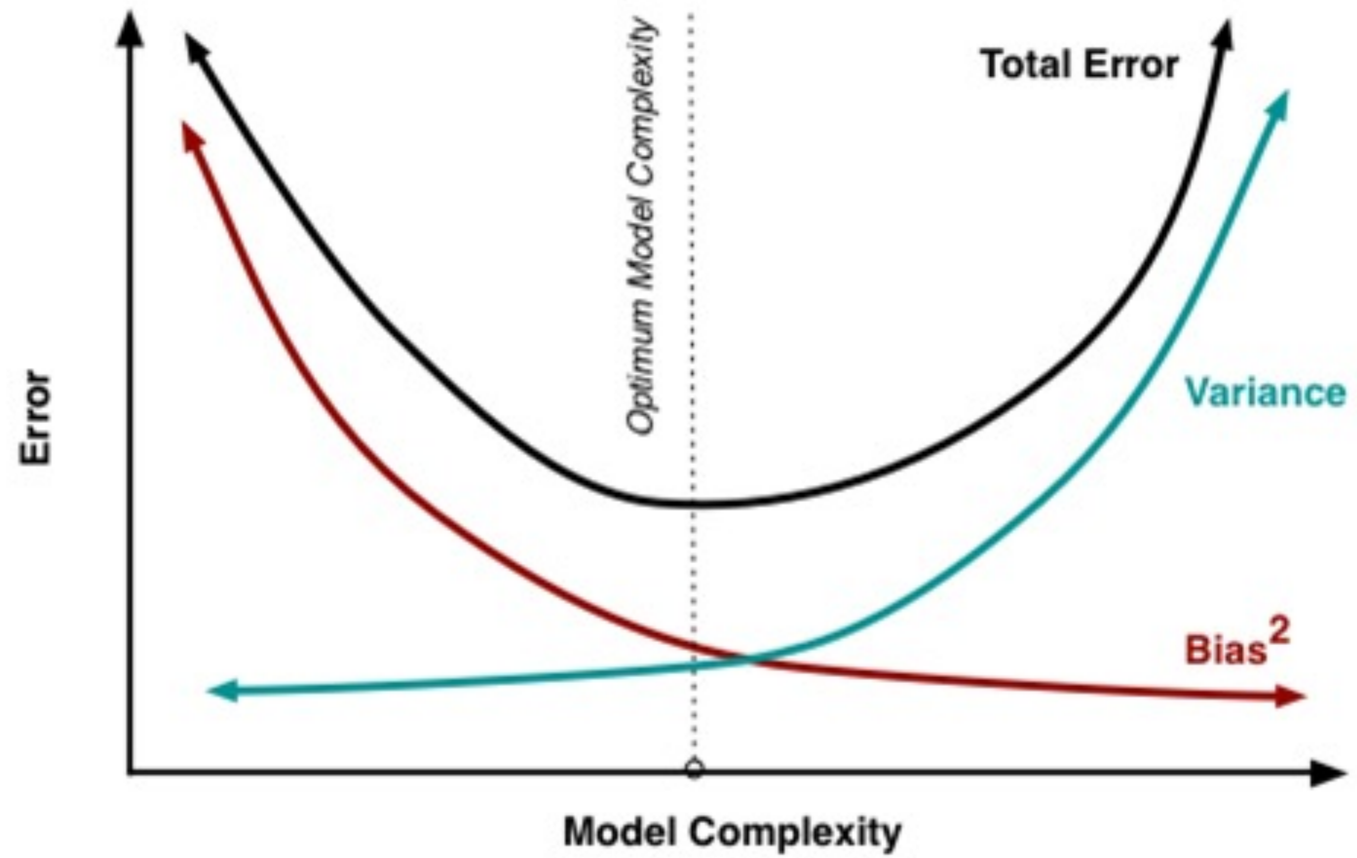
- **Bias** is the inability of a model to learn enough about the relationship between the predictors **X** and the response **Y.** It quantifies how much on an average the predicted values differ from the actual value

- **Variance** quantifies a model's tendency to learn *too much* about the relationship that's implied by the training dataset. It represents a model's lack of consistency across different datasets

$$\text{total error} = \text{irreducible error} + \underbrace{\text{error due to bias} + \text{error due to variance}}_{\text{reducible error}}$$

MLDS Center
Maryland Longitudinal Data System

Bias - Variance Tradeoff



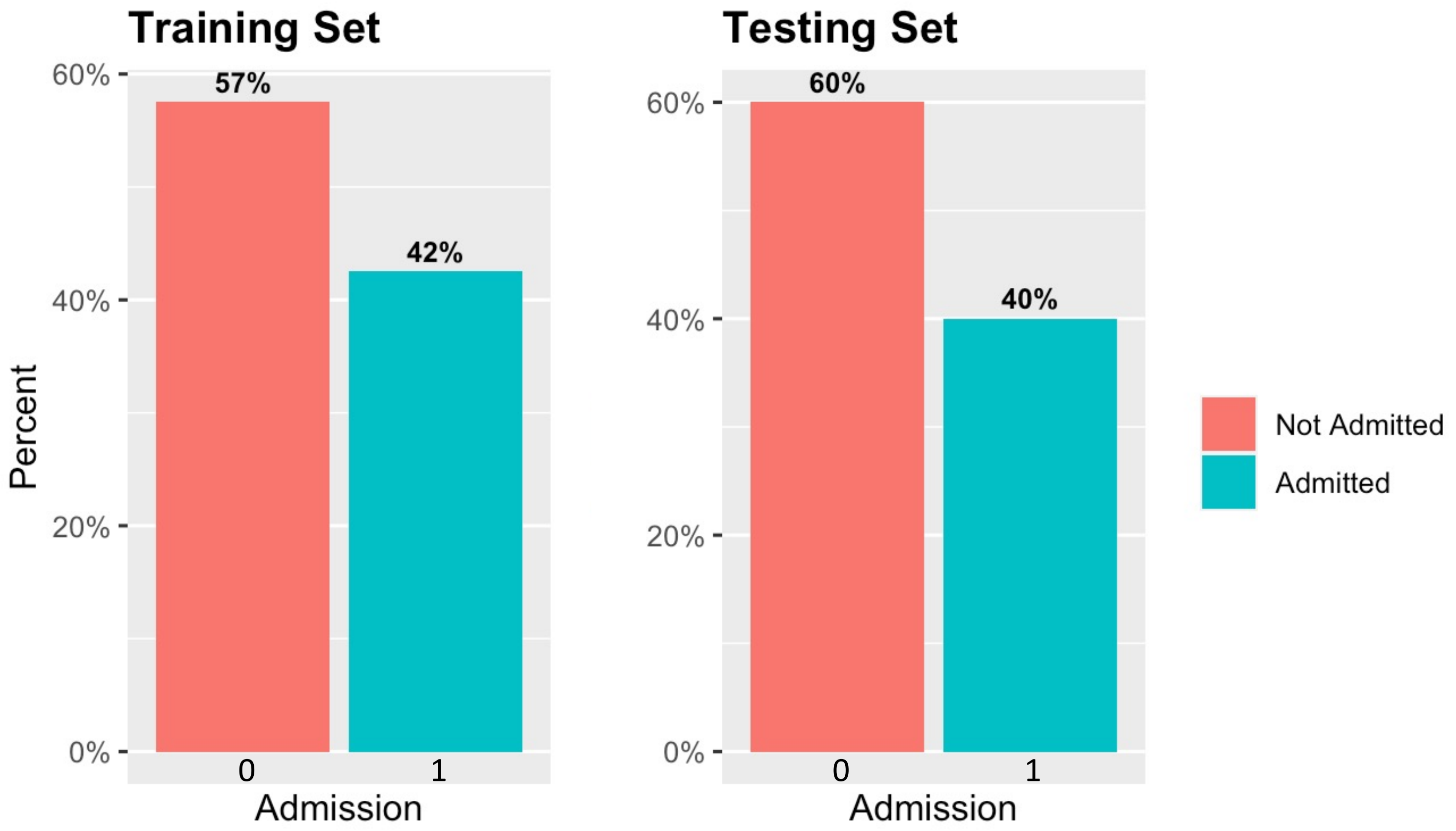
Some Common Methods

Machine Learning Algorithms

Characteristics	ML Algorithm	Tuning
Without dimension reduction	**Modal Classification** Multiple Linear Regression **Logistic regression** k-Nearest Neighbor (kNN)	None None None Number of neighbors
Dimension reduction with penalty	**Lasso**	Shrinkage/ penalty
Tree based, non-linear relationship	**Classification/**Regression **Trees** **Random forest**	Tree depth/ pruning Number and depth of trees
Non-linear decision surface	Support vector machine Neural network	Kernels Depth of neurons
Ensemble of many algorithms	Super learner (SL)	Weights

Modal Classification for Grad Admission

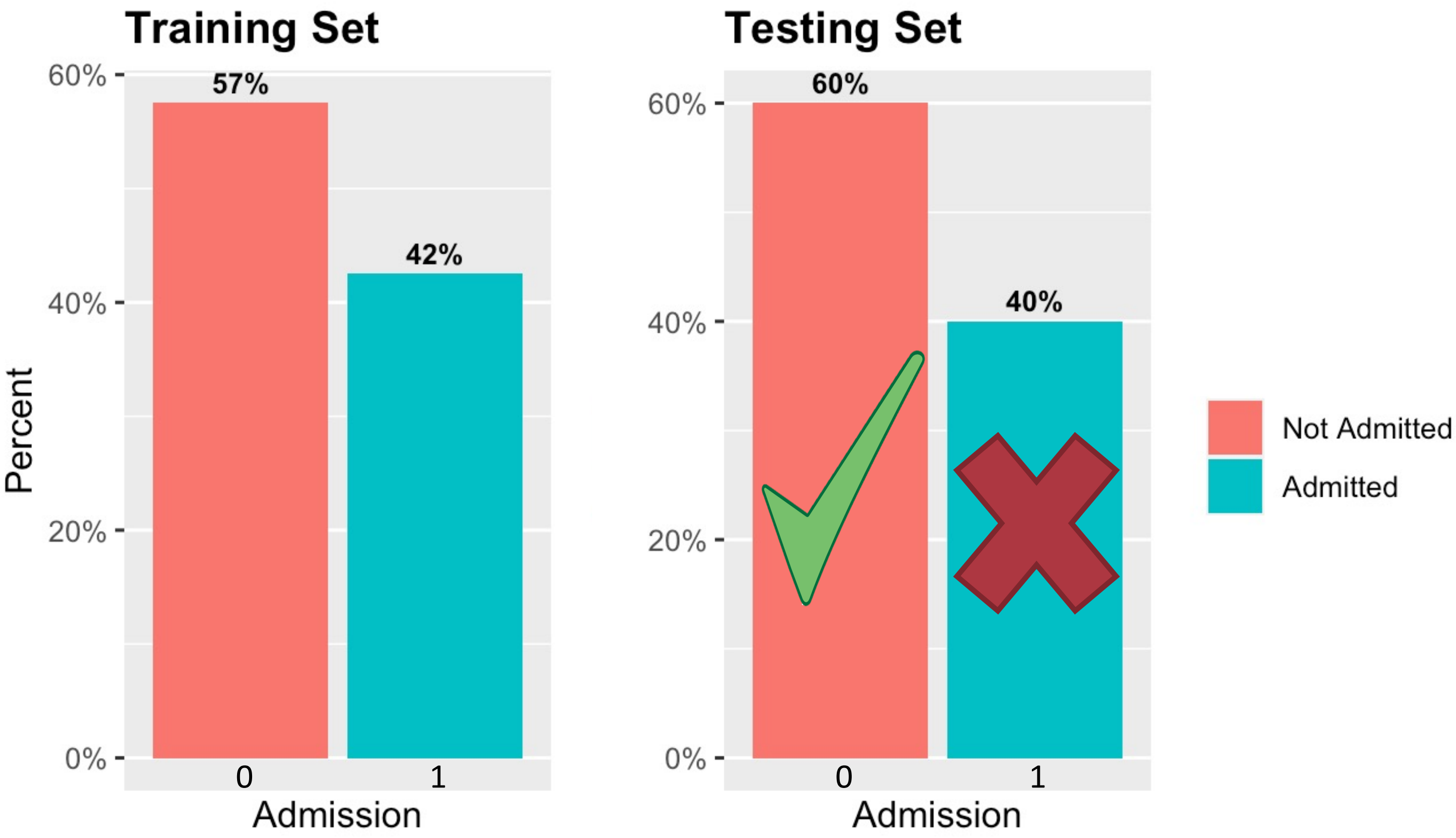
- Baseline Measure for Comparison
- Majority Rule



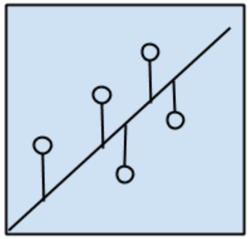
MLDS Center
Maryland Longitudinal Data System

Modal Classification for Grad Admission

- 60% Accuracy



Logistic Regression



Regression Algorithms

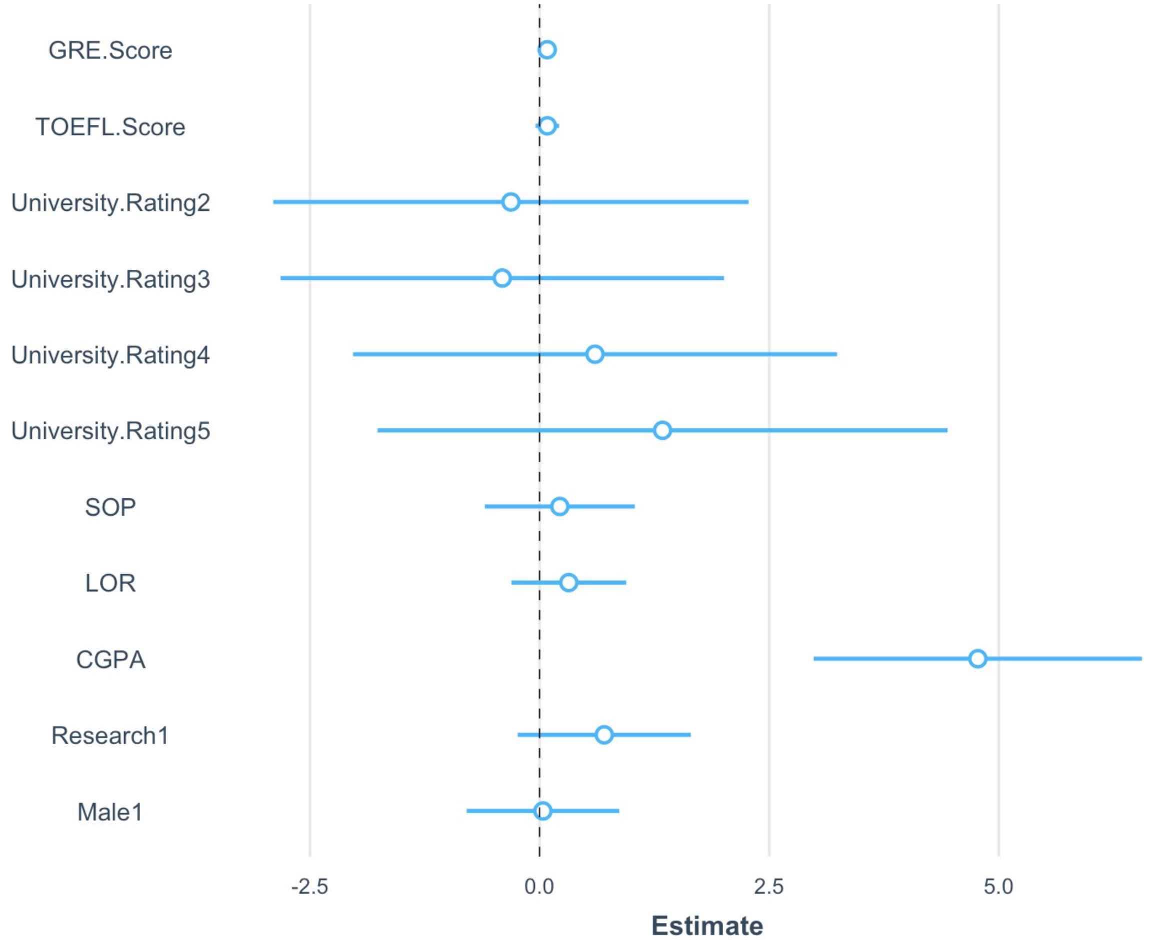
- The outcome of interest is a dichotomous variable
  - Predictions are made using the formula:

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

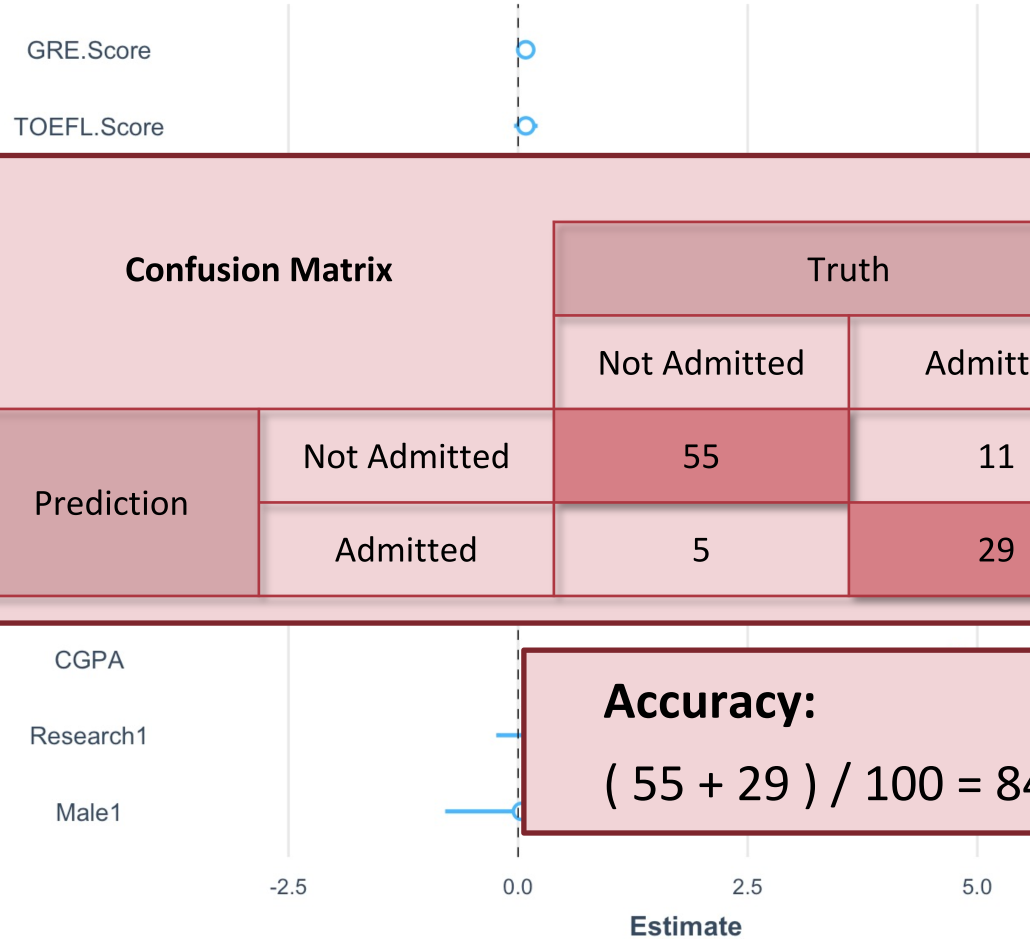
$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}$$

- Can be generalized to more than two classes by using a linear function for each class
- A simple approach to supervised learning but assumes linearity (which often isn't the truth)
- Linear models are easy to interpret

Logistic Regression for Grad Admission



Logistic Regression for Grad Admission



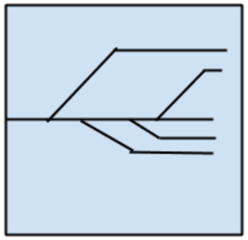
Confusion Matrix		Truth	
		Not Admitted	Admitted
Prediction	Not Admitted	55	11
	Admitted	5	29

Accuracy:

(55 + 29) / 100 = 84%

MLDS Center
Maryland Longitudinal Data System

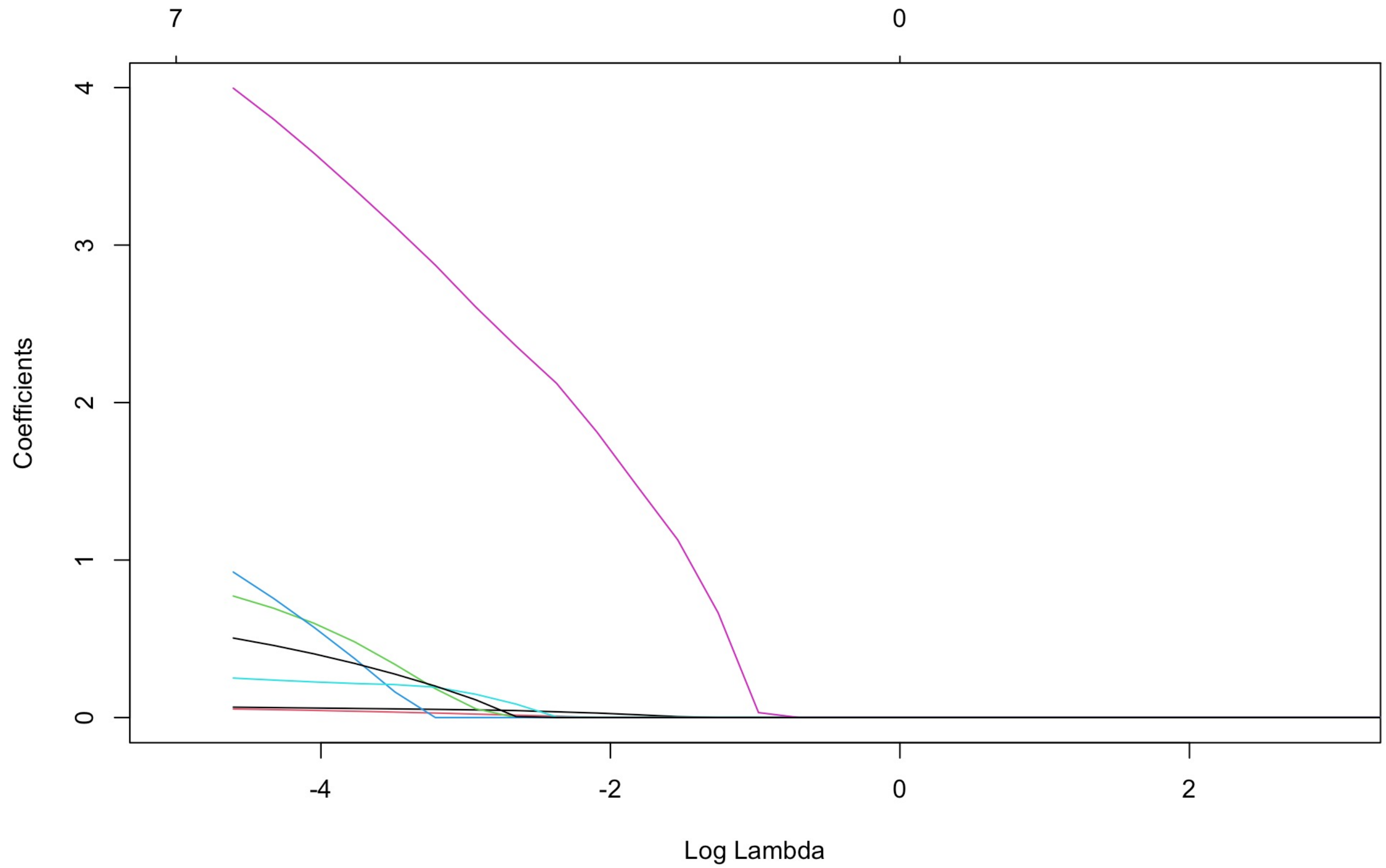
Lasso Regression


Regularization Algorithms

- Variable selection method that shrinks the coefficient estimates towards zero based on a penalty (tuning) parameter λ
- Selecting a good value of λ for the lasso is critical; cross-validation is again the method of choice
- Produces a model that can include only a subset of the predictor variables which reduces the model complexity and helps avoid over-fitting to the data

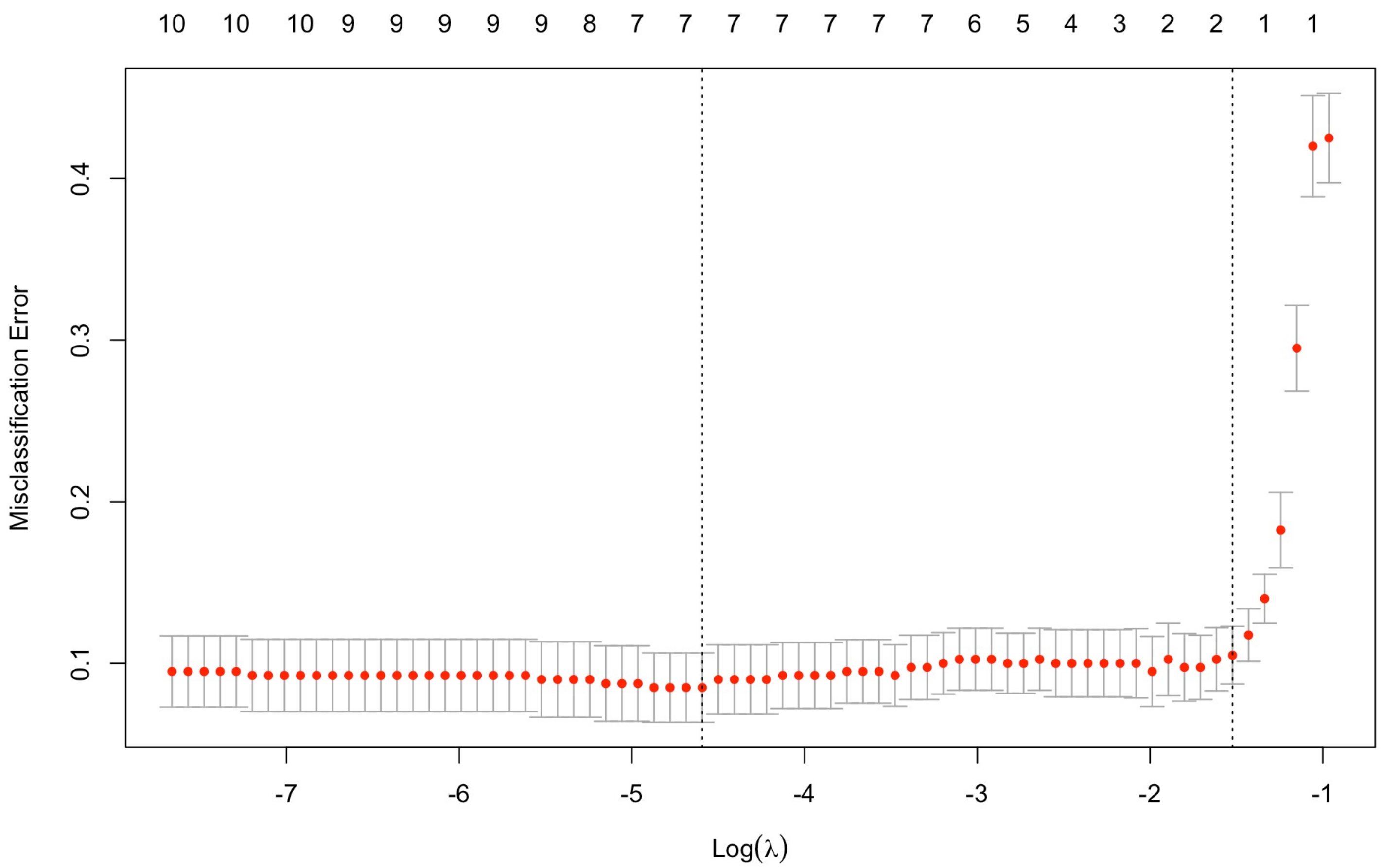
MLDS Center
Maryland Longitudinal Data System

Lasso for Grad Admission



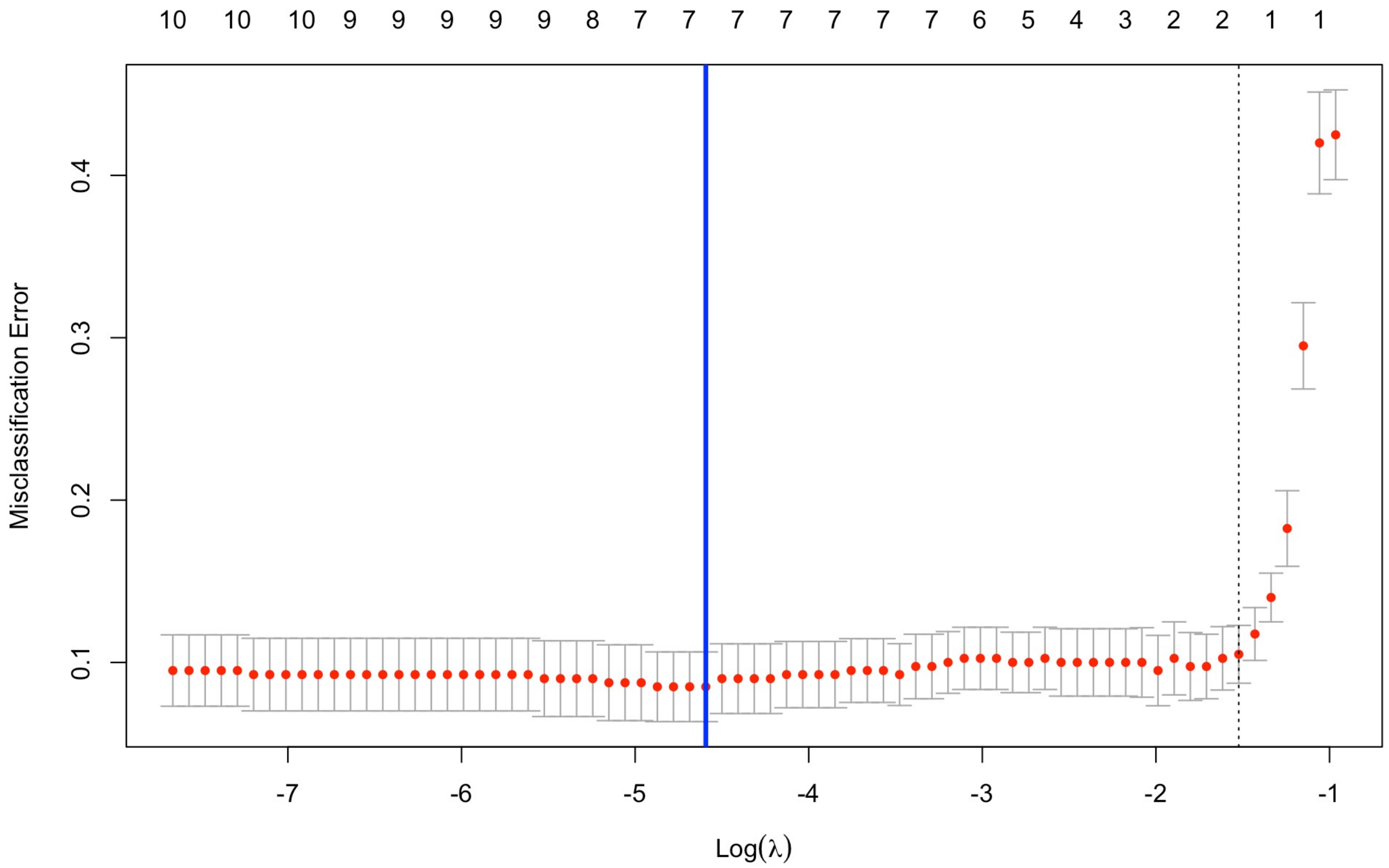
MLDS Center
Maryland Longitudinal Data System

Lasso for Grad Admission

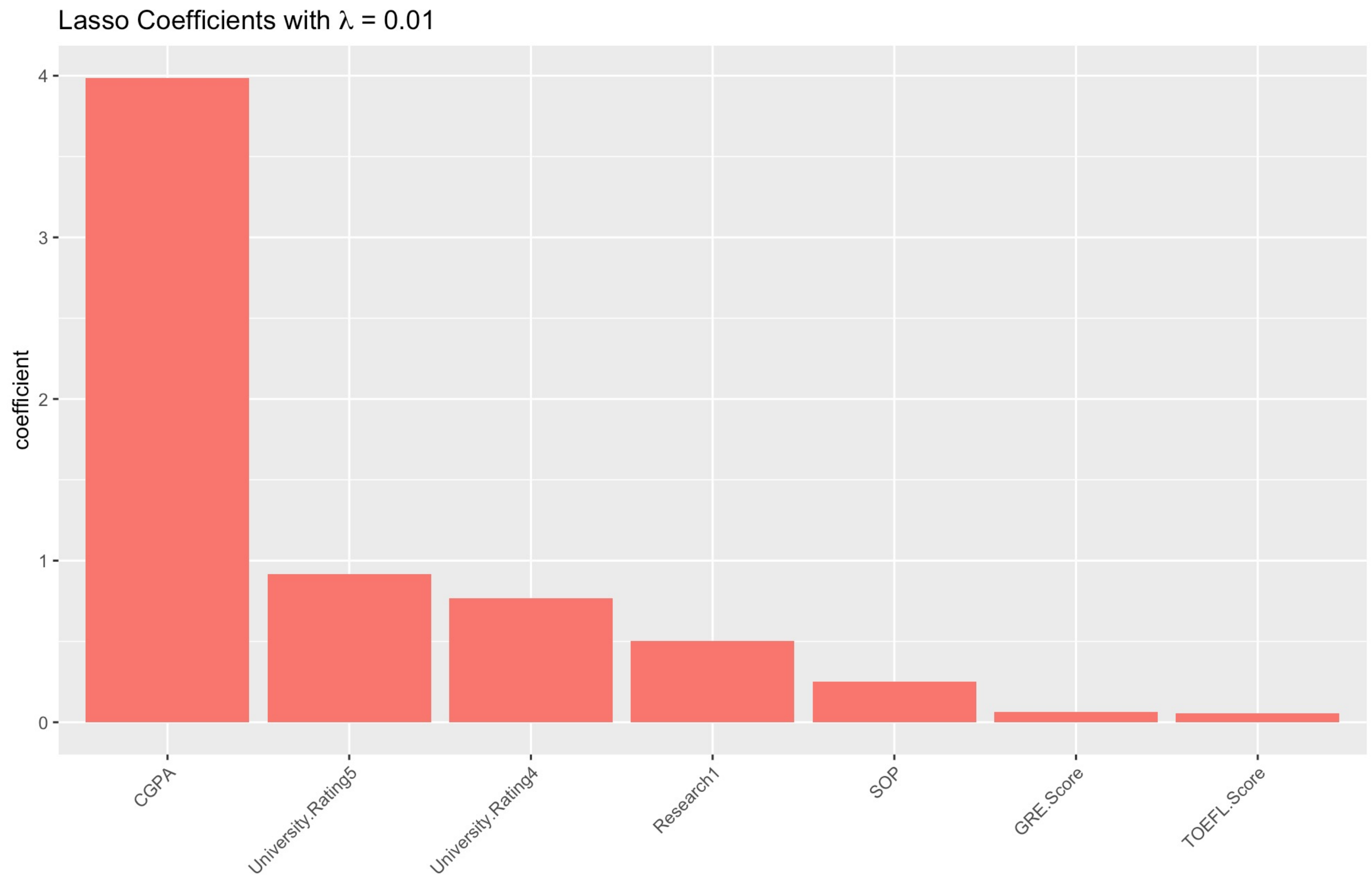


MLDS Center
Maryland Longitudinal Data System

Lasso for Grad Admission

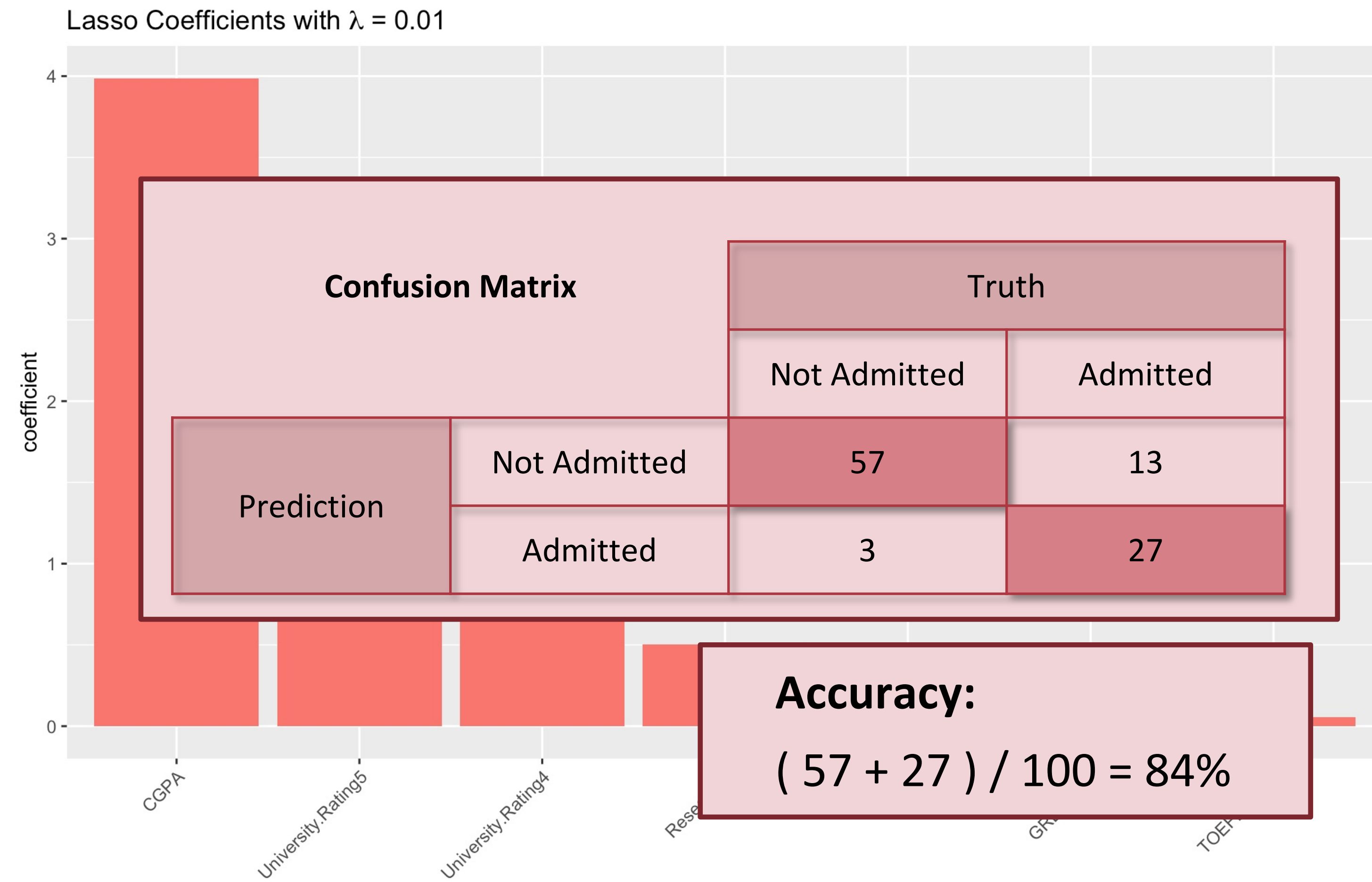


Lasso for Grad Admission



MLDS Center
Maryland Longitudinal Data System

Lasso for Grad Admission

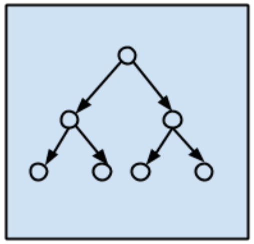


Confusion Matrix		Truth	
		Not Admitted	Admitted
Prediction	Not Admitted	57	13
	Admitted	3	27

Accuracy:

(57 + 27) / 100 = 84%

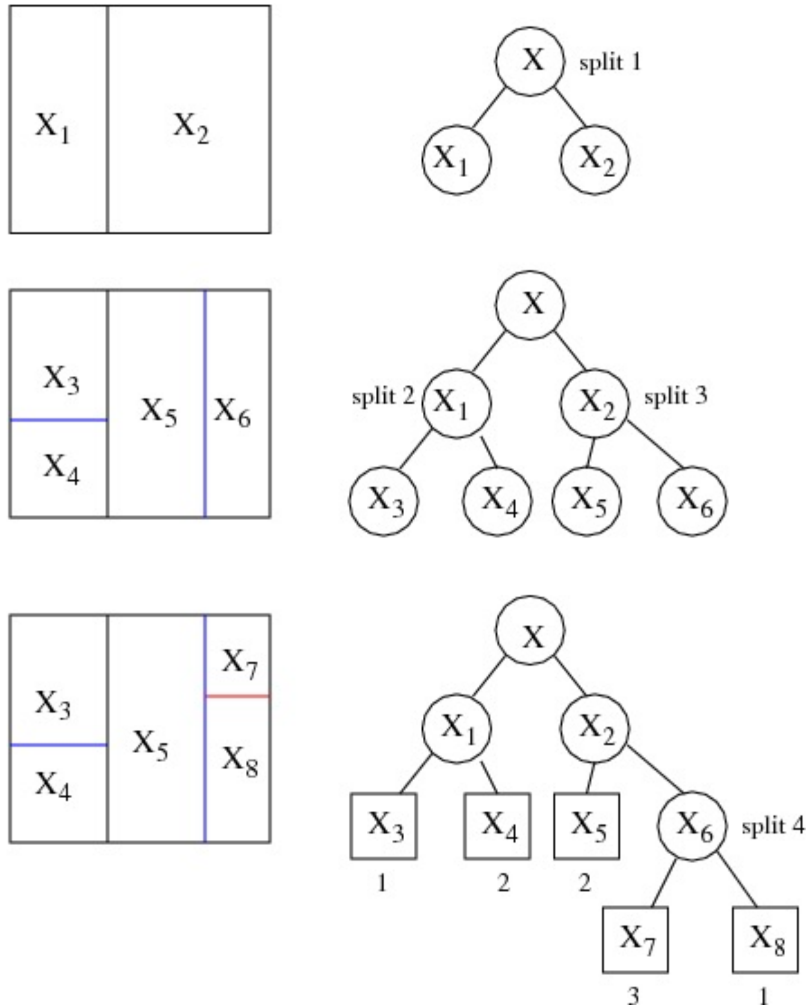
Decision Trees

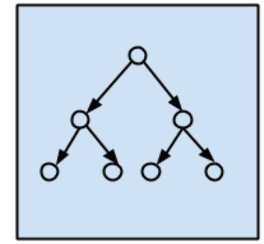


Decision Tree
Algorithms

- *Classification* or *Regression*
- Nonparametric models built in the form of a tree structure by stratifying or segmenting the predictor space into several simple regions
- Complexity (tuning) Parameter α
- Within each final node, the predicted value is either the modal value/class of the outcome (Classification) or the mean of the outcome variable for observations in the node (Regression)
- Easy to interpret

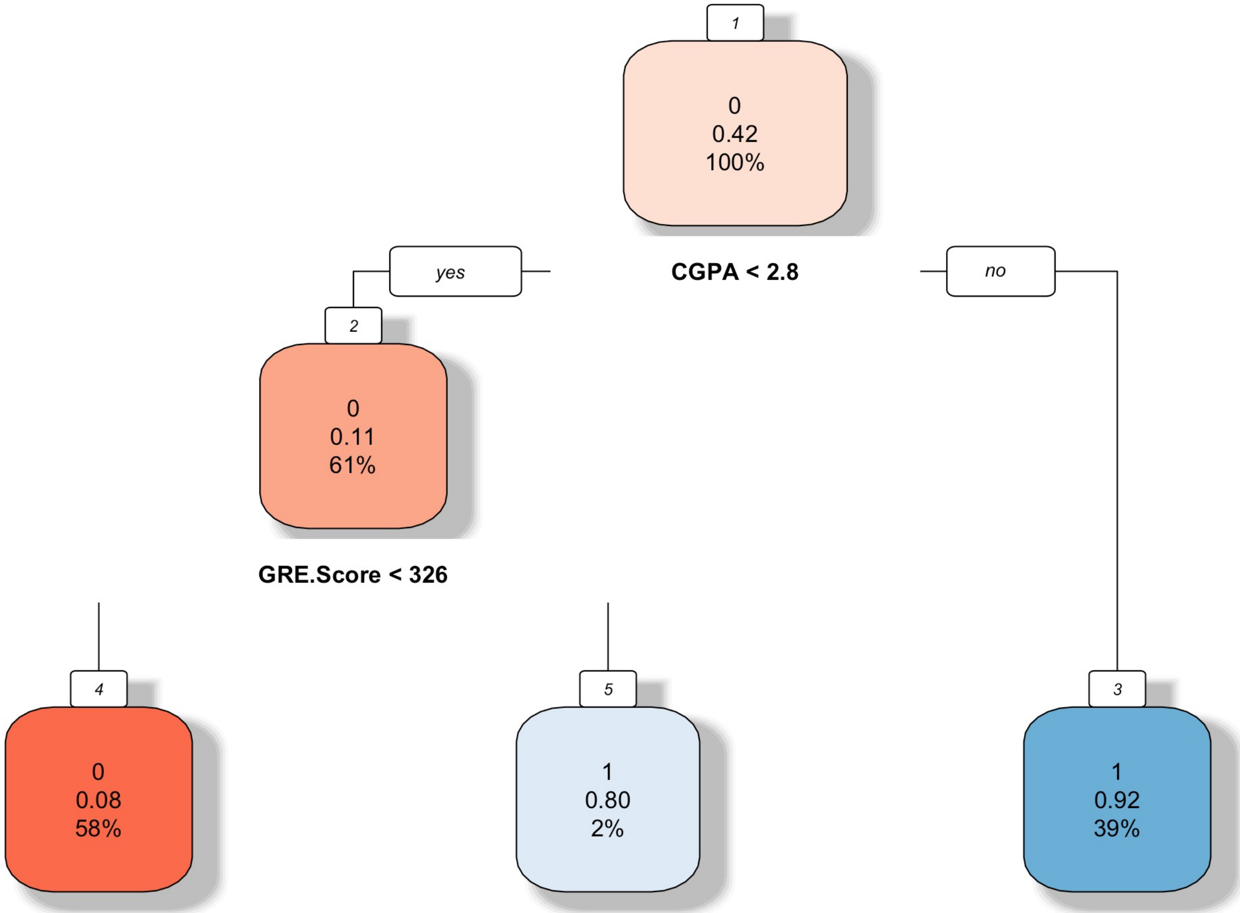
Decision Tree Process





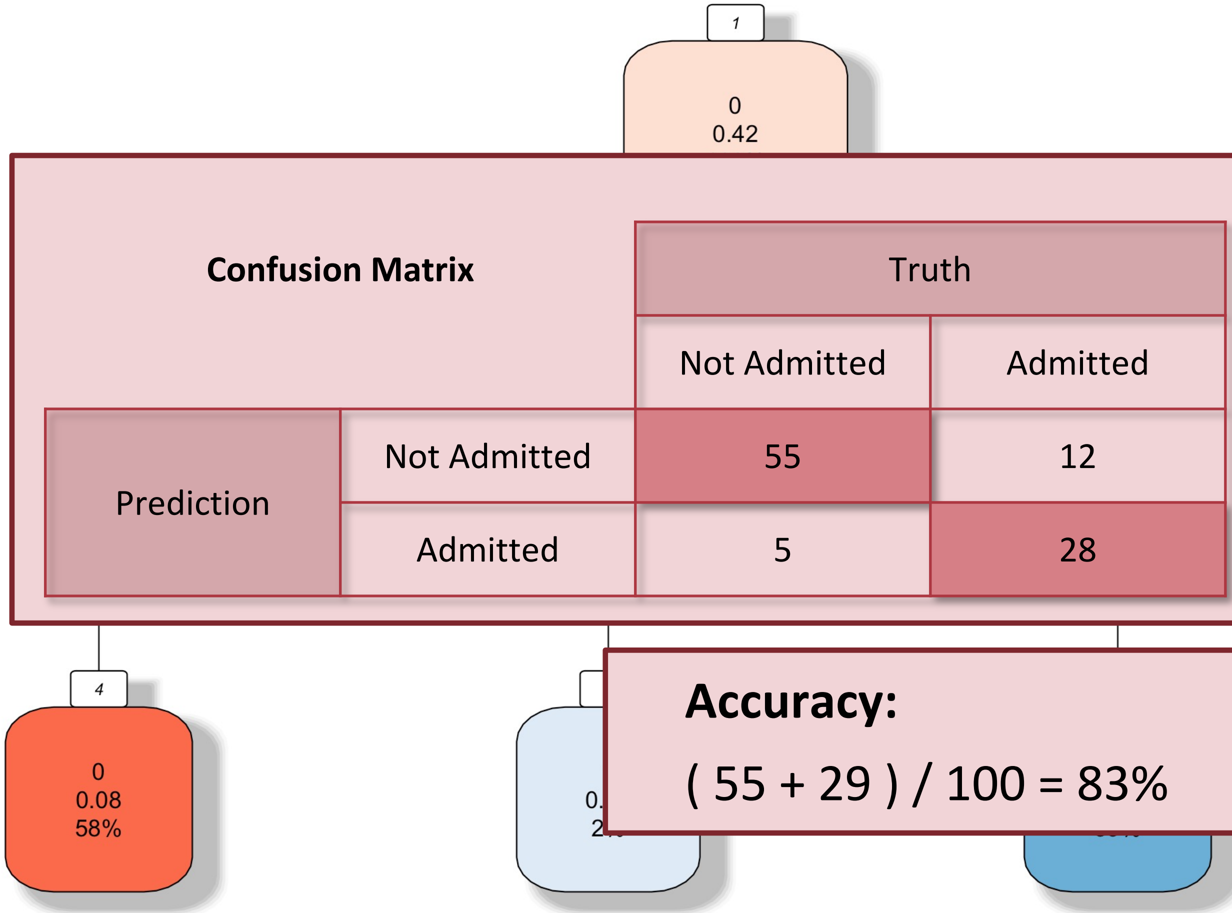
Decision Tree Algorithms

Decision Tree for Grad Admission



MLDS CENTER
Maryland Longitudinal Data System

Decision Tree for Grad Admission

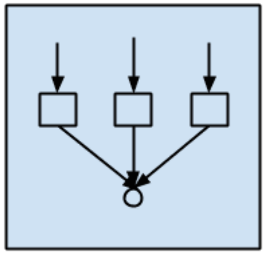


Confusion Matrix		Truth	
		Not Admitted	Admitted
Prediction	Not Admitted	55	12
	Admitted	5	28

Accuracy:

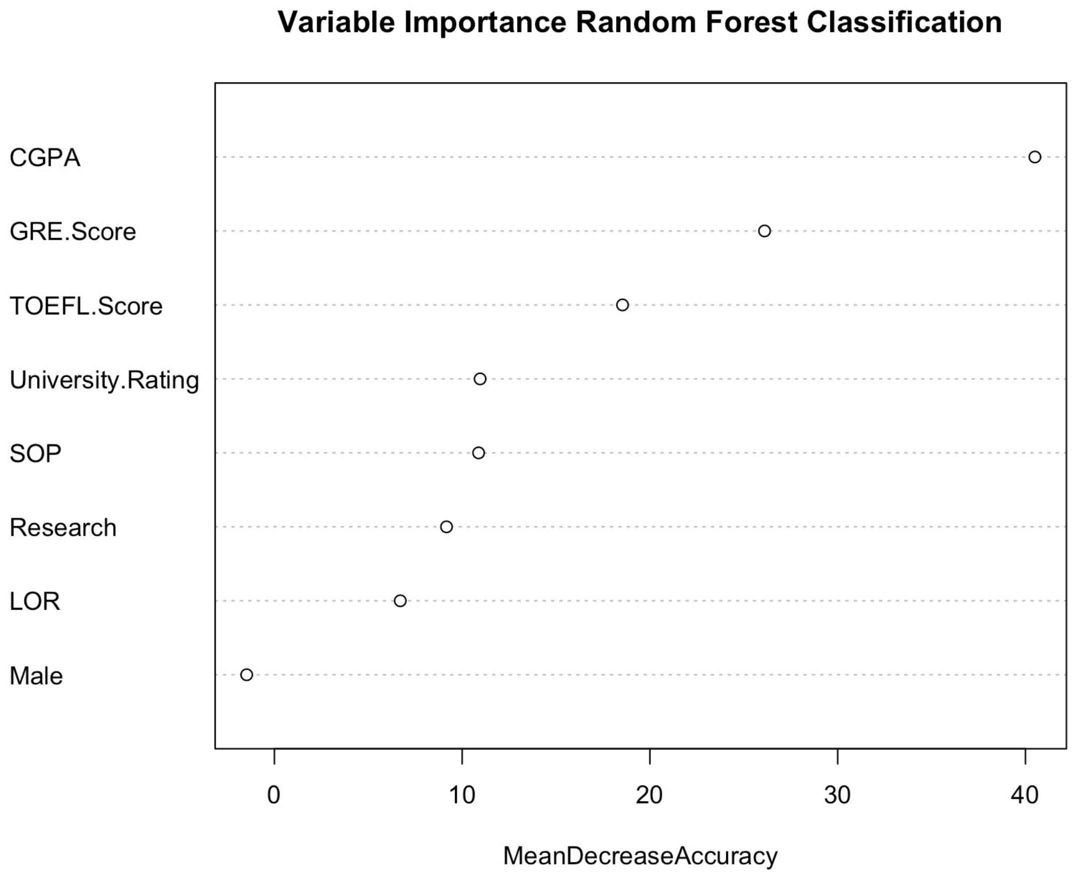
(55 + 29) / 100 = 83%

Random Forest


Ensemble Algorithms

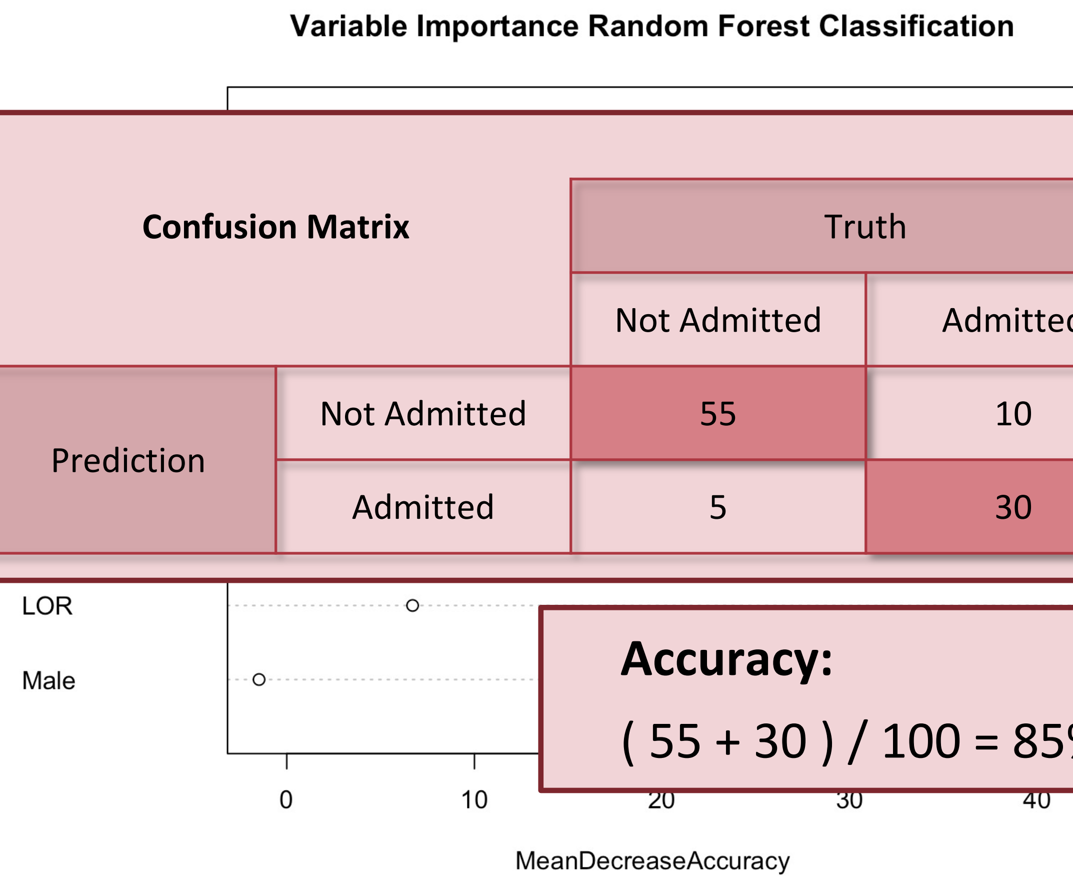
- Used for *Classification* or *Regression*
- An *ensemble* classifier which combines the results of many decision tree models built on bootstrapped samples using a random sample of the predictors at each split
  - A selection of m predictors is taken at each split (typically $m \approx \sqrt{p}$)
- This process decorrelates the trees which reduces the variance
- Need to select the number of trees

Random Forest for Grad Admission



MLDS Center
Maryland Longitudinal Data System

Random Forest for Grad Admission



Confusion Matrix		Truth	
		Not Admitted	Admitted
Prediction	Not Admitted	55	10
	Admitted	5	30

Accuracy:

(55 + 30) / 100 = 85%

MLDS Application

Future Data Science Projects

- **ML Prediction**
- Can we reasonably predict student success variables?
- Do machine learning algorithms more accurately predict these outcomes over other methods?

- **For What Purpose?**
- Missing data: Absent students and/or years where certain assessments were not used
- To examine the effects of local school system or state policies on student success

Creating a Dataset

All students in Grade A

- Identifying a particular cohort (Grade A in Year 2016)

Students who took Test X

- All students who took the test in previous years

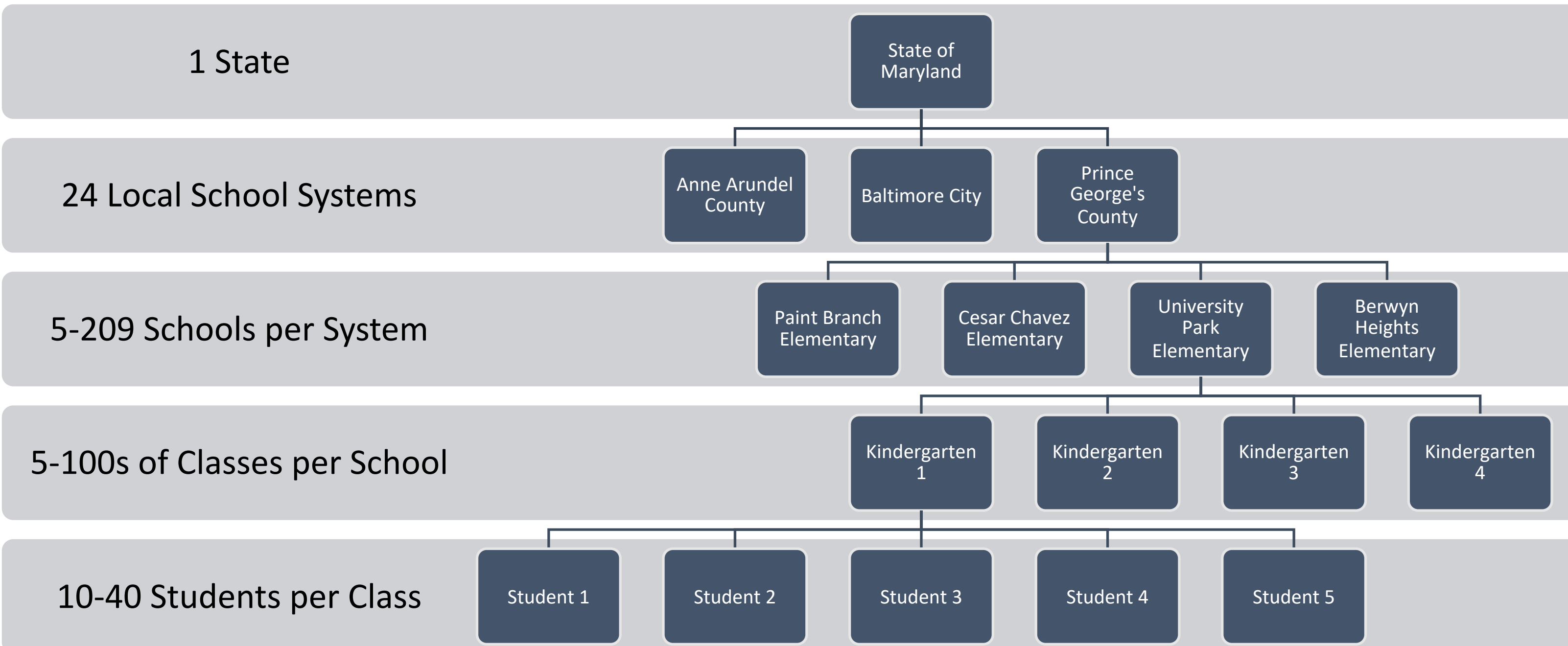
Identifying the test score

- Which score to use?

Selecting covariates

- Missingness vs representation

Nested Data is Common in Education



Nested Data is Common in Education

1 State

24 Local School Systems

5-209 Schools per System

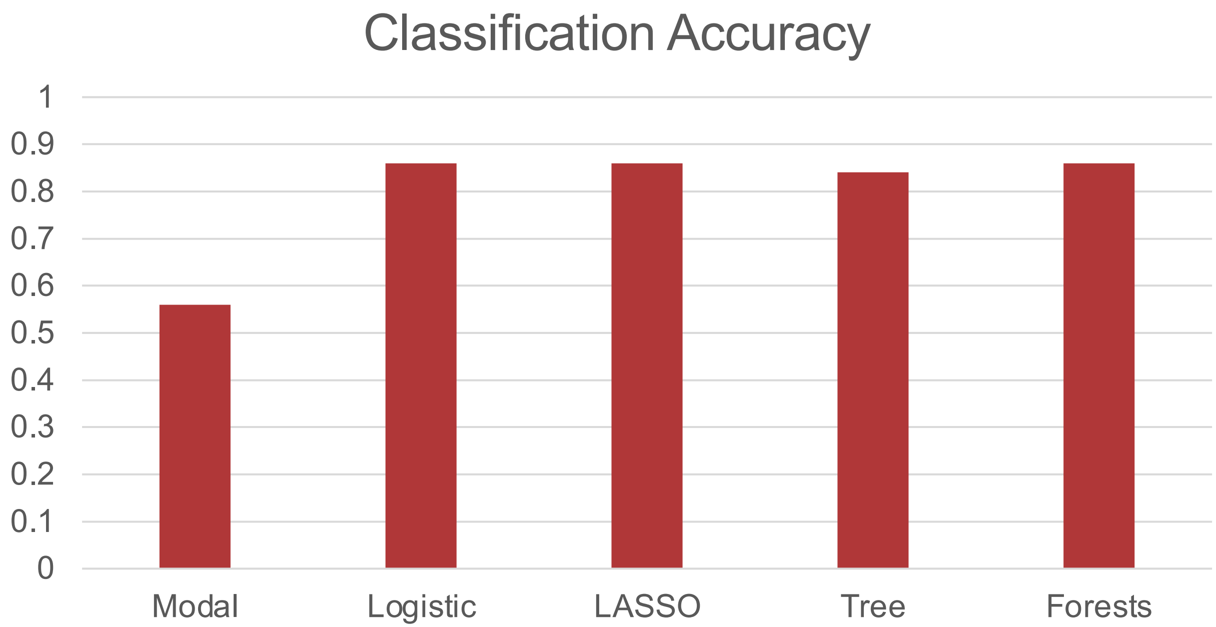
5-100s of Classes per School

10-40 Students per Class

Student

- With existing MLDS classroom, school, and local school system covariates, can the nested data structure be ignored?
- Do ML algorithms do better than parametric models in terms of prediction accuracy?

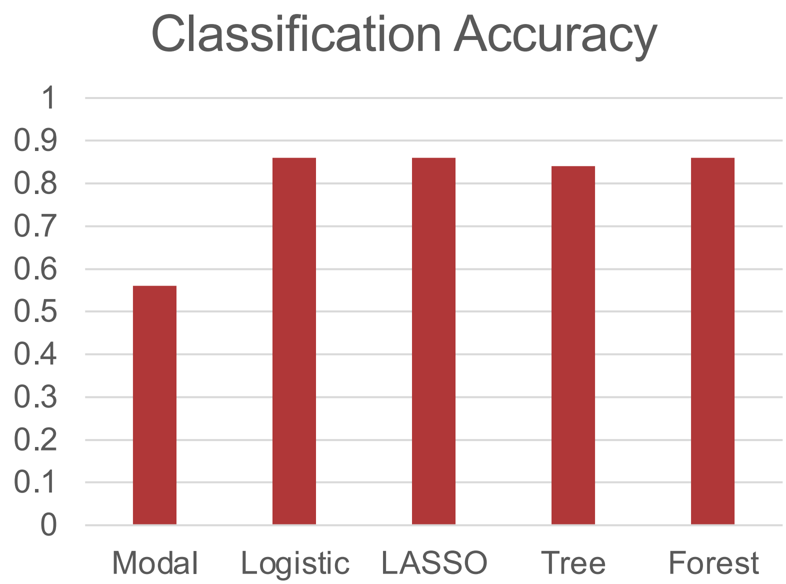
Some Preliminary Results



MLDS Center
Maryland Longitudinal Data System

Considerations When Applying to MLDS Data

- Are all types of students being classified equally well?
- Which groups of students are being classified better? Which groups are worse?
  - Race
  - Gender
  - Grade
  - ELL
  - FARMS
  - Local School System
- Does this vary by method?



MLDS Center
Maryland Longitudinal Data System

Current Data Science Project

- Which algorithm is accurately predicting which types of students?
- Is there a way to leverage high accuracy across all groups?
- Are these algorithms better than parametric models (multilevel logistic regression)?
- If we can accurately predict student outcomes, how can and should these predictions be used to support students?

Thank you!
Questions?

- Tracy Sweet; tsweet@umd.edu
- Brennan Register; brr@umd.edu
- Patrick Sheehan; psheehan@umd.edu