TEXAS A&M UNIVERSITY
Engineering

ECEN 758 Data Mining and Analysis: Lecture 13, Decision Tree Classification

Joshua Peeples, Ph.D.
Assistant Professor
Department of Electrical and Computer Engineering

Announcements

TEXAS A&M UNIVERSITY Engineering

- Assignment #3 will be released next Wednesday (10/09)
- No class next Monday (10/07): Fall Break
- Guest lecture next Wednesday (10/09)
  - +1 on Midterm exam for attendance (must sign-in to attendance sheet)
  - +3 for one paragraph (5 – 7 sentences) summary of presentation (must attend lecture)
  - Section 700: Watch recording and submit 1 paragraph (+1) or 2 paragraphs (+3)
    - Can also attend lecture virtually and submit 1 paragraph
- Exam I on Monday, 10/14

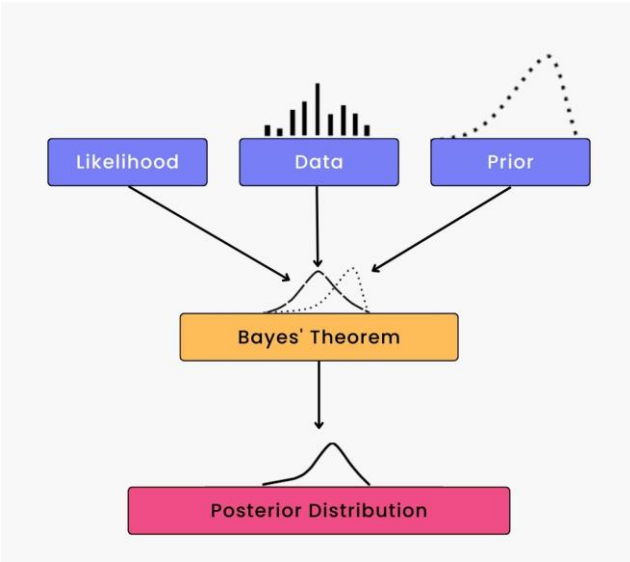
Midterm Overview

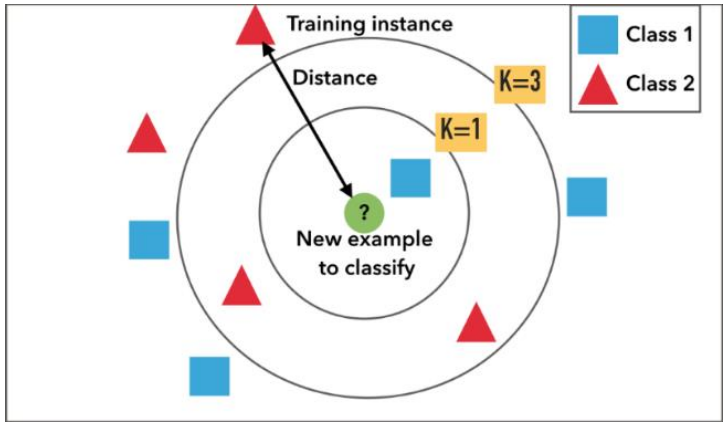
TEXAS A&M UNIVERSITY Engineering

- Midterm format
  - 40 questions
  - Multiple choice/answer, T/F
- In class exam and closed book (no coding IDEs, e.g., Jupyter Notebooks, Google Colab, etc.)
  - Online students will use Honorlock
  - Exam will be administered through Canvas
- Allowed to bring calculator
- One sheet of notes (two-sided, 8.5x11 inches)
  - Cannot be electronic copy
- I will bring scratch sheets of paper to class (if needed)
- Exam will cover lectures 2 – 13
- Focus on Data Mining/ML fundamentals discussed in lecture material, assignments, and textbooks

Last Lecture

- Introduction to Classification I
  - Definition
  - Model types
  - Hyperparameters vs Parameters
- Bayesian and Nearest Neighbor Classification





Today

TEXAS A&M UNIVERSITY
Engineering

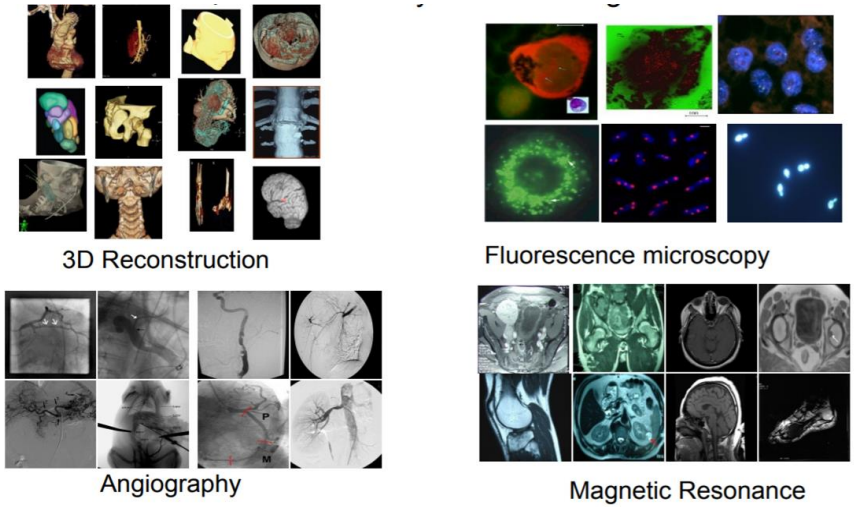
- Introduction to Classification II
  - Methodology
  - Metrics
  - Overfitting and underfitting
- Decision tree classification
- Reading: ZM Chapter 18
- Supplemental: ZM Chapter 22

Classification Overview

Classification Tasks

TEXAS A&M UNIVERSITY Engineering

- Classification: given features X, predict label (class) y
- Examples:
  - Communication symbol recognition (input signal w/source & channel induced noise, classes: valid communication symbols)
  - Medical diagnosis (input: symptoms, classes: diseases)
  - Character recognition (input: handwritten characters, classes: {a..z})
  - Fraud detection (input: account activity, classes: fraud / no fraud)
  - … many more



Src: Nowka 2015, IBM CLEF 2013

Image from: Madhu Ramiah- Medium

Classification Methodology

Supervised Learning

TEXAS A&M UNIVERSITY
Engineering

Learning from experience

0: Macaw 1: Conure

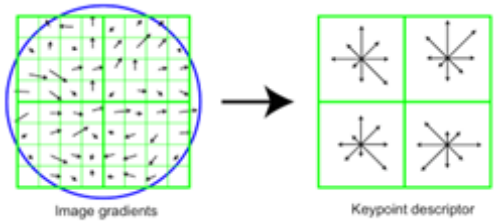
Supervised Learning

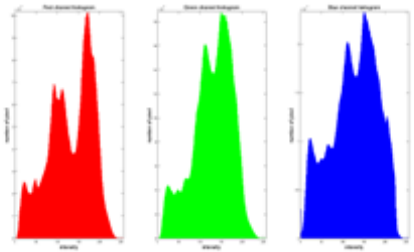
TEXAS A&M UNIVERSITY Engineering

Training:

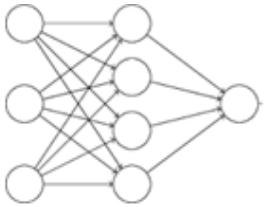
Collect Labeled Training Data

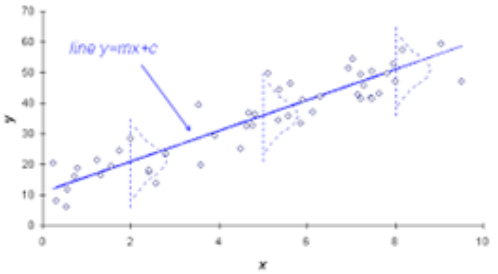
Extract Features



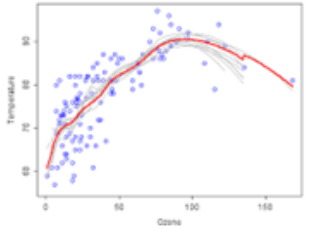


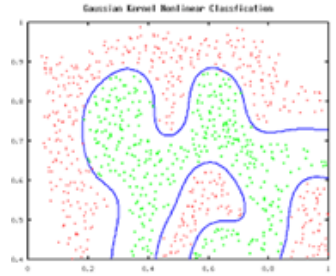
Select a Model





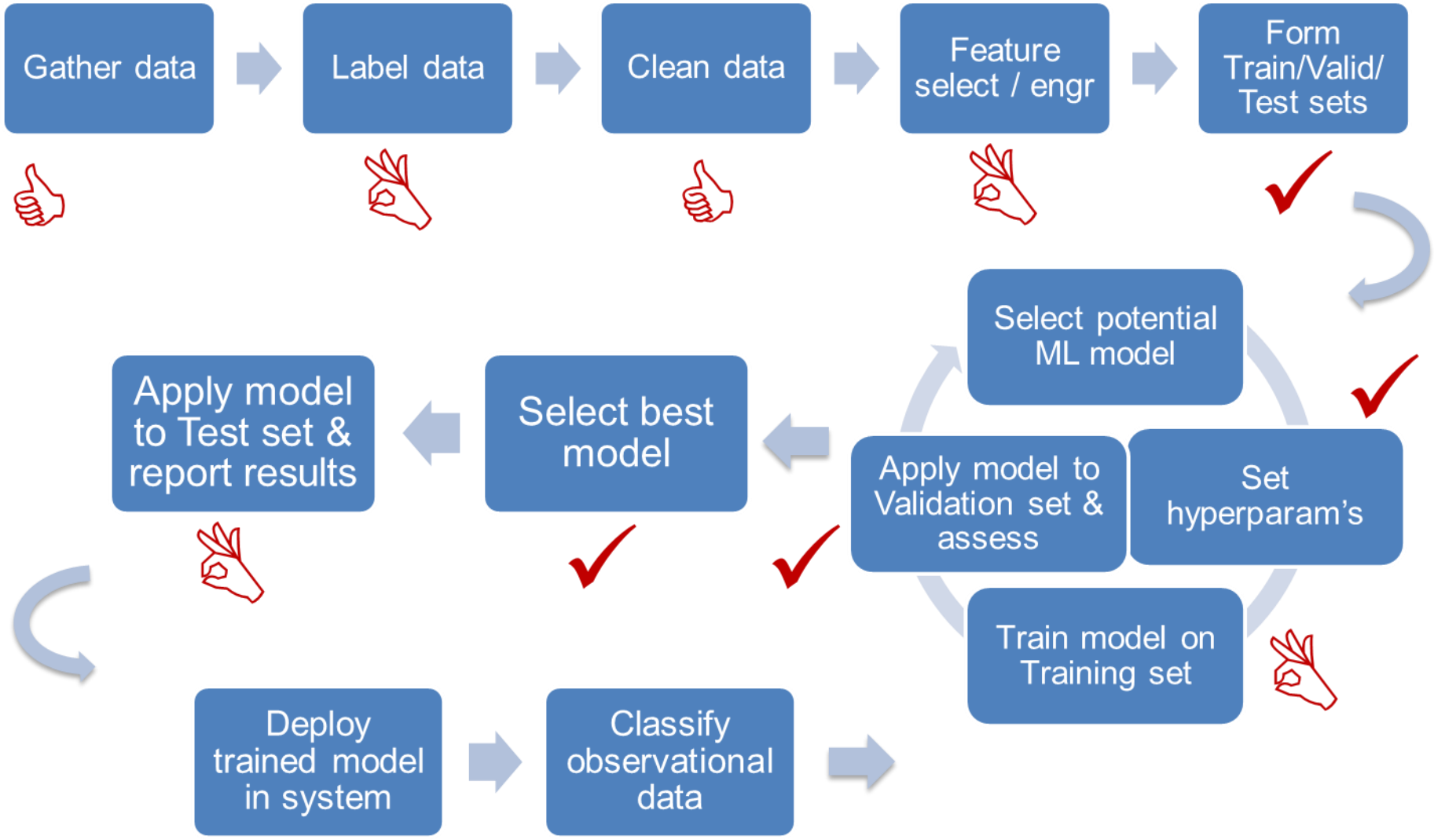
Fit the Model



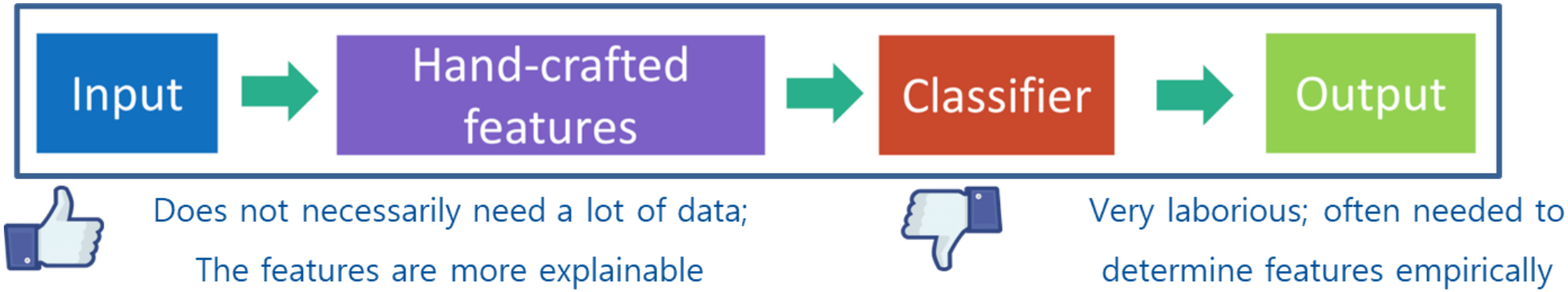


General Supervised Learning Process

TEXAS A&M UNIVERSITY
Engineering



Classic ML Approach

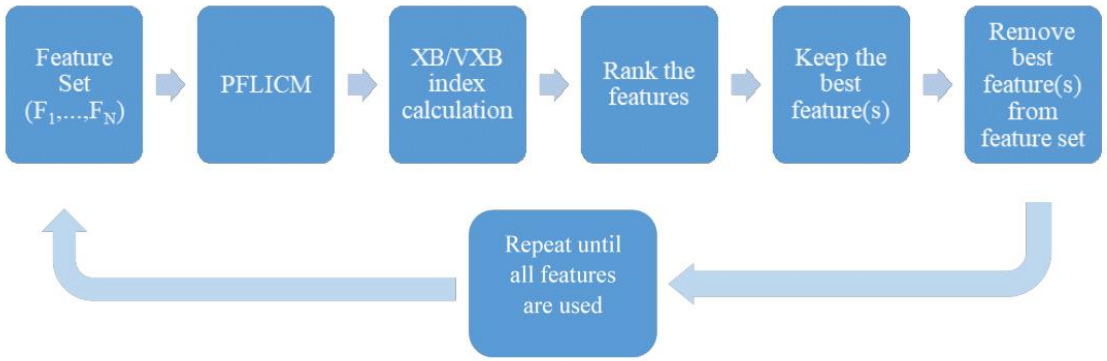


- Feature engineering
  - Feature extraction
  - Feature selection

Feature Selection

TEXAS A&M UNIVERSITY
Engineering

1) Explicitly decide which features to use in training of your model.
   - Examine correlations to aid in selecting
   - Some techniques like Random Forest provide feedback on relative feature importance
2) And/or use a method that drops features with least value in fitting/testing
3) Feature selection techniques:
   a) Wrapper
   b) Filter
   c) Embedded

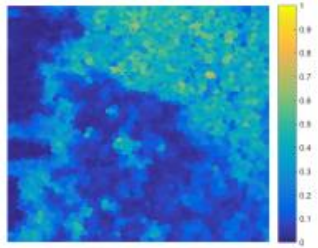


J. Peeples, et al. Possibilistic fuzzy local information C-means with automated feature selection for seafloor segmentation. SPIE, 2018.

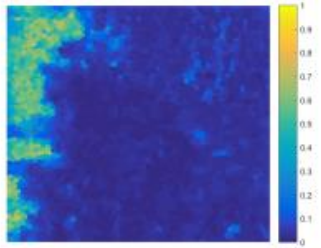
Feature Selection: Wrapper Method

TEXAS A&M UNIVERSITY
Engineering

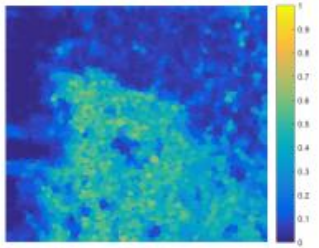
(a) Original Image



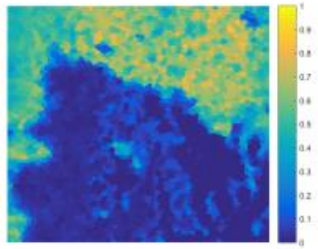
(b) PFLICM Cluster 1



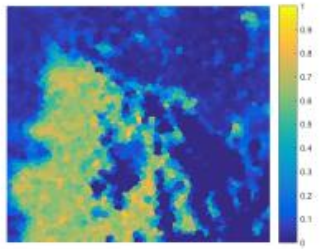
(c) PFLICM Cluster 2



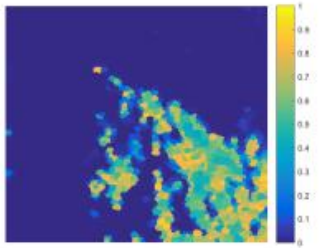
(d) PFLICM Cluster 3



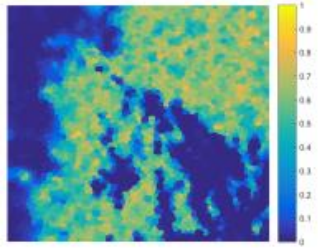
(e) PFLICM Cluster 1



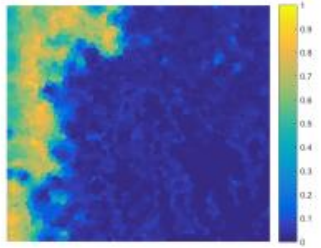
(f) PFLICM Cluster 2



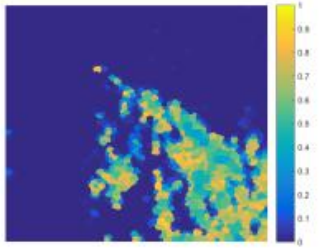
(g) PFLICM Cluster 3



(h) PFLICM Cluster 1



(i) PFLICM Cluster 2



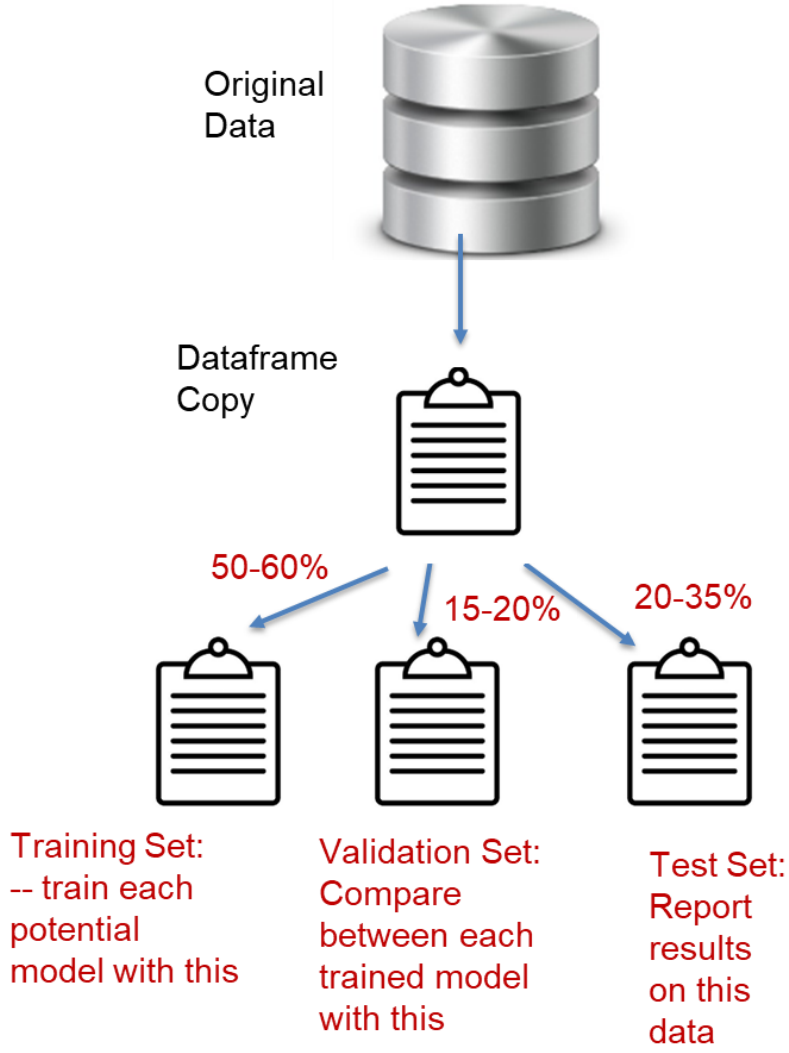
(j) PFLICM Cluster 3

J. Peeples, et al. Possibilistic fuzzy local information C-means with automated feature selection for seafloor segmentation. SPIE, 2018.

Data Splitting

TEXAS A&M UNIVERSITY
Engineering

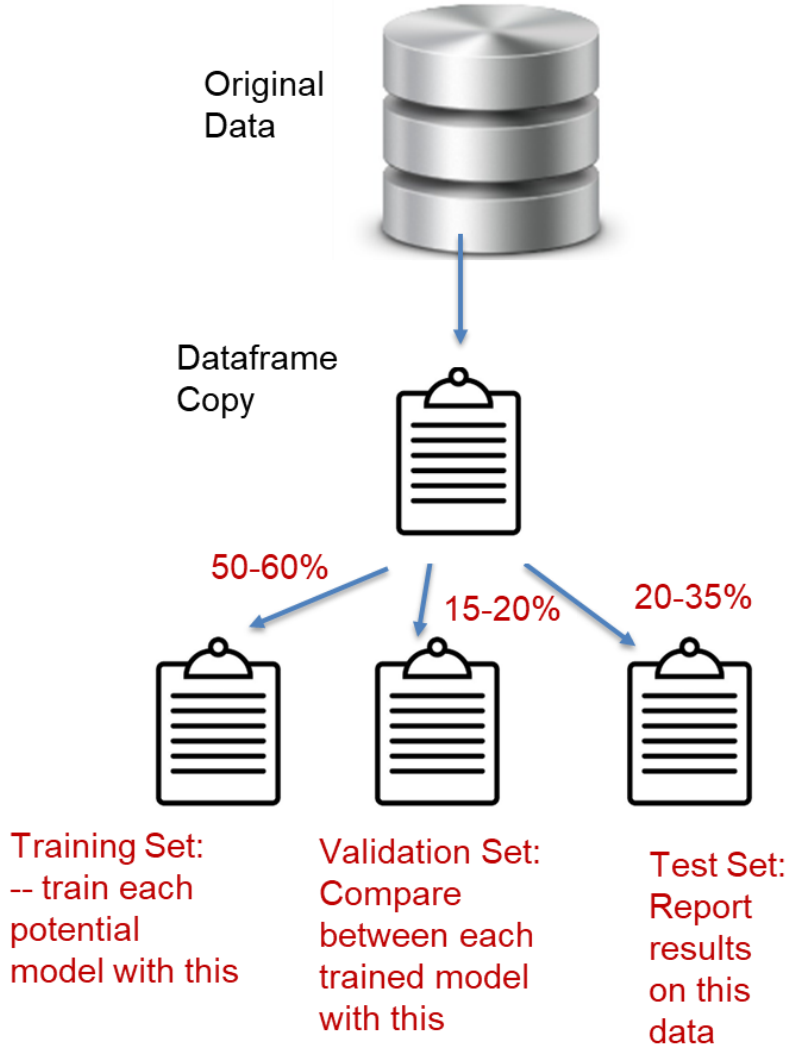
1. **Never contaminate original data** – clean, extract, feature engineer on a copy (or better, on a "dataframe" in an ML tool)



Data Splitting

TEXAS A&M UNIVERSITY Engineering

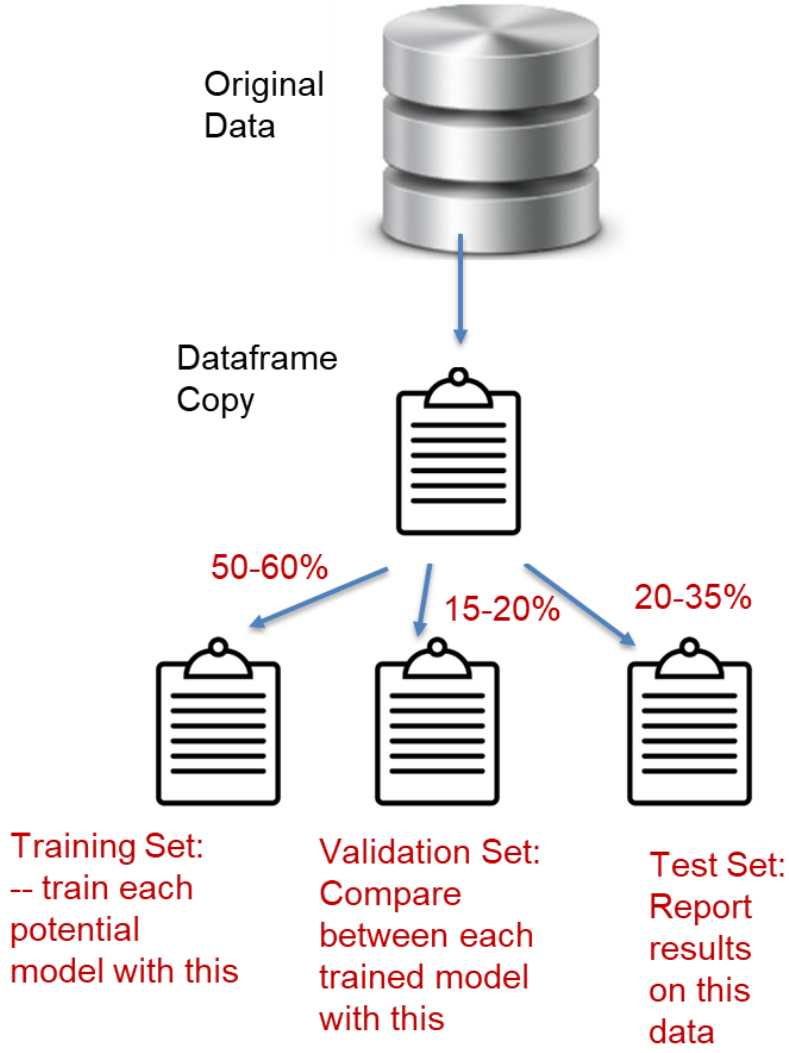
1. **Never contaminate original data** – clean, extract, feature engineer on a copy (or better, on a "dataframe" in an ML tool)
2. **Never evaluate the ML model on data used to train the model**
   i. Objective is to see how well the model does on new data. Keep training and test datasets separate



Data Splitting

TEXAS A&M UNIVERSITY Engineering

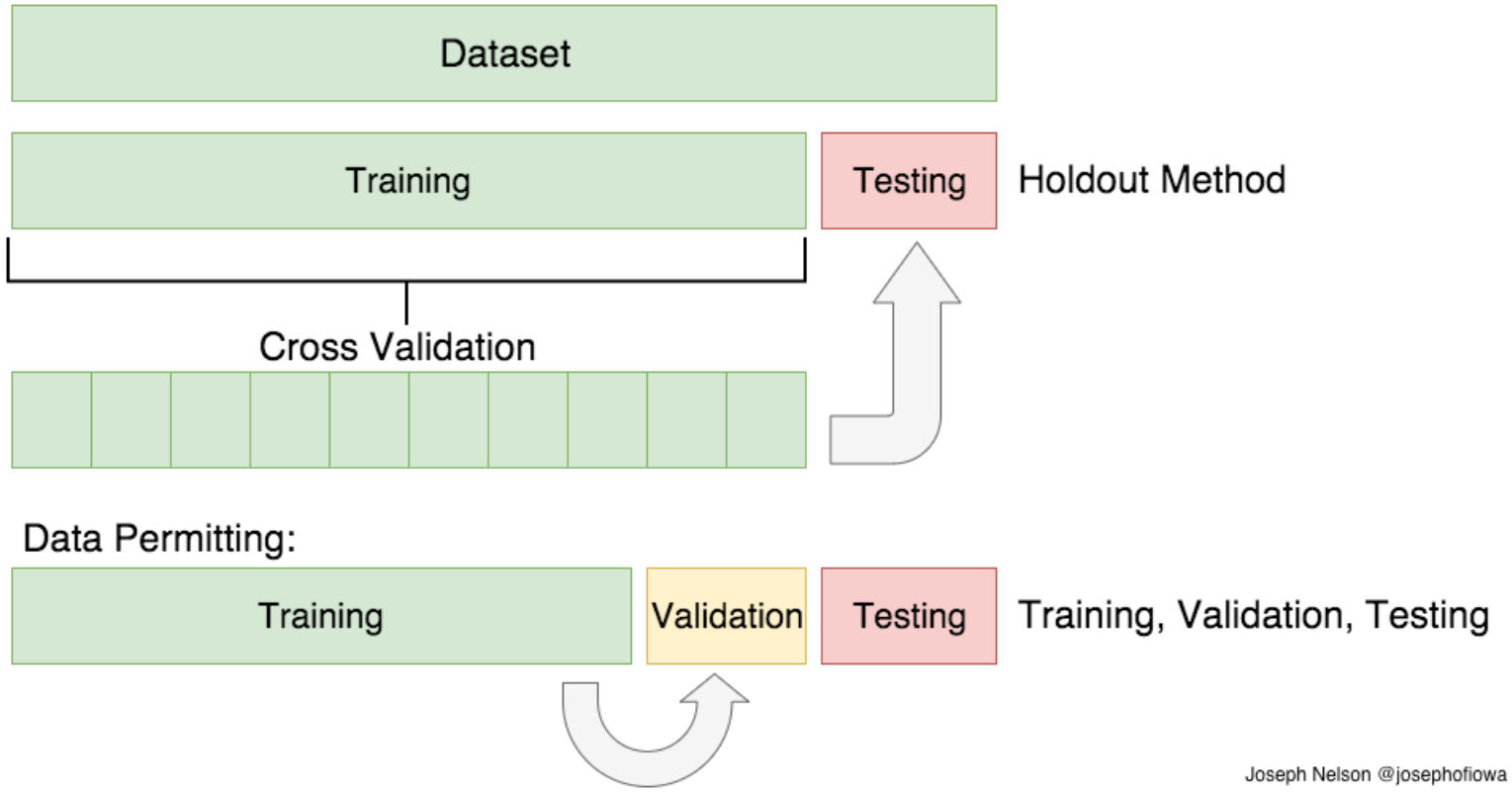
1. **Never contaminate original data** – clean, extract, feature engineer on a copy (or better, on a "dataframe" in an ML tool)
2. **Never evaluate the ML model on data used to train the model**
   i. Objective is to see how well the model does on new data. Keep training and test datasets separate
3. **Never make selection between alternative models/settings on data used to train or test the model**
   i. Keep a separate validation dataset for this



Train/Validation/Test datasets

TEXAS A&M UNIVERSITY
Engineering

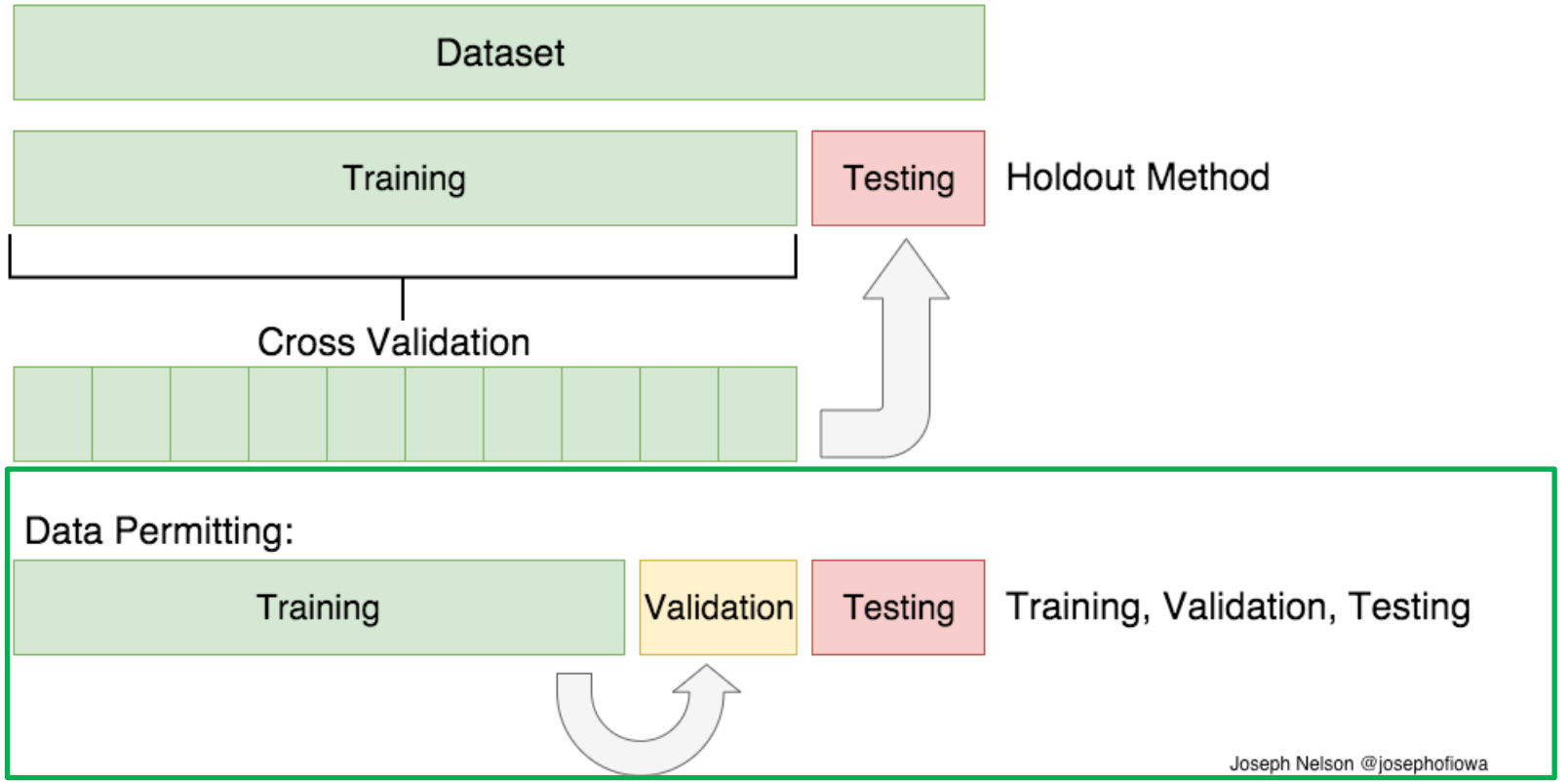
- Split data into train, validation, and test



Train/Validation/Test datasets

TEXAS A&M UNIVERSITY Engineering

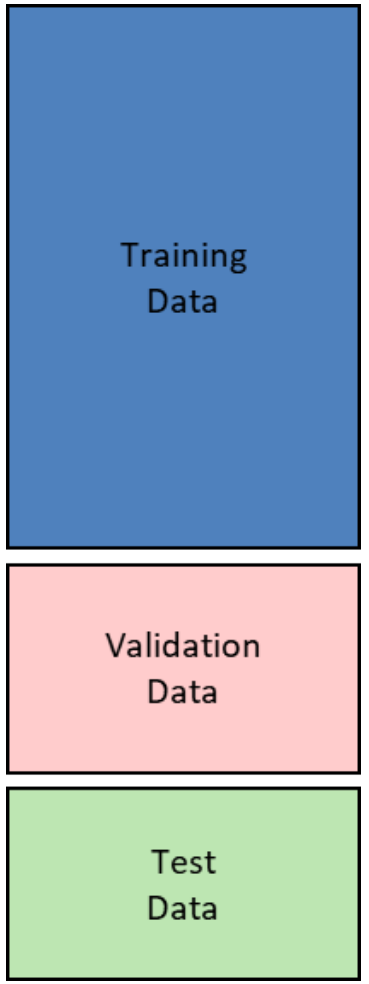
- Split data into train, validation, and test



Train/Validation/Test Splits

TEXAS A&M UNIVERSITY
Engineering

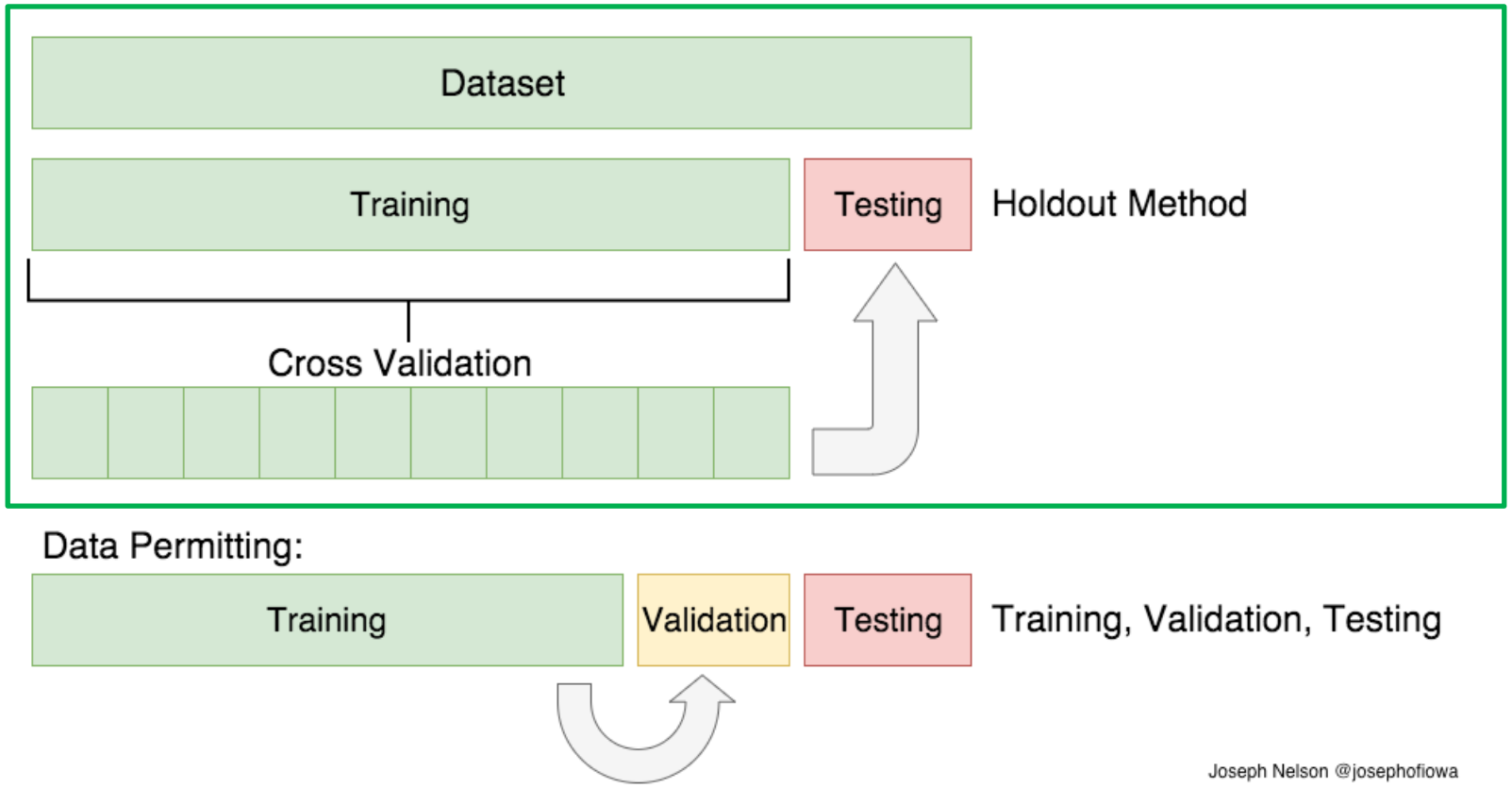
- Data: labeled instances
  - Training set (typically 50-80%)
  - Validation set (typically 10-20%)
  - Test set (typically 10-30%)
- Experimentation cycle
  - The training set is used to fit the models
  - The validation set is used to determine which hyperparameters result in the lowest estimated prediction error for model selection
  - The test set is used for assessment of the generalization error of the final chosen model



Train/Validation/Test datasets

TEXAS A&M UNIVERSITY Engineering

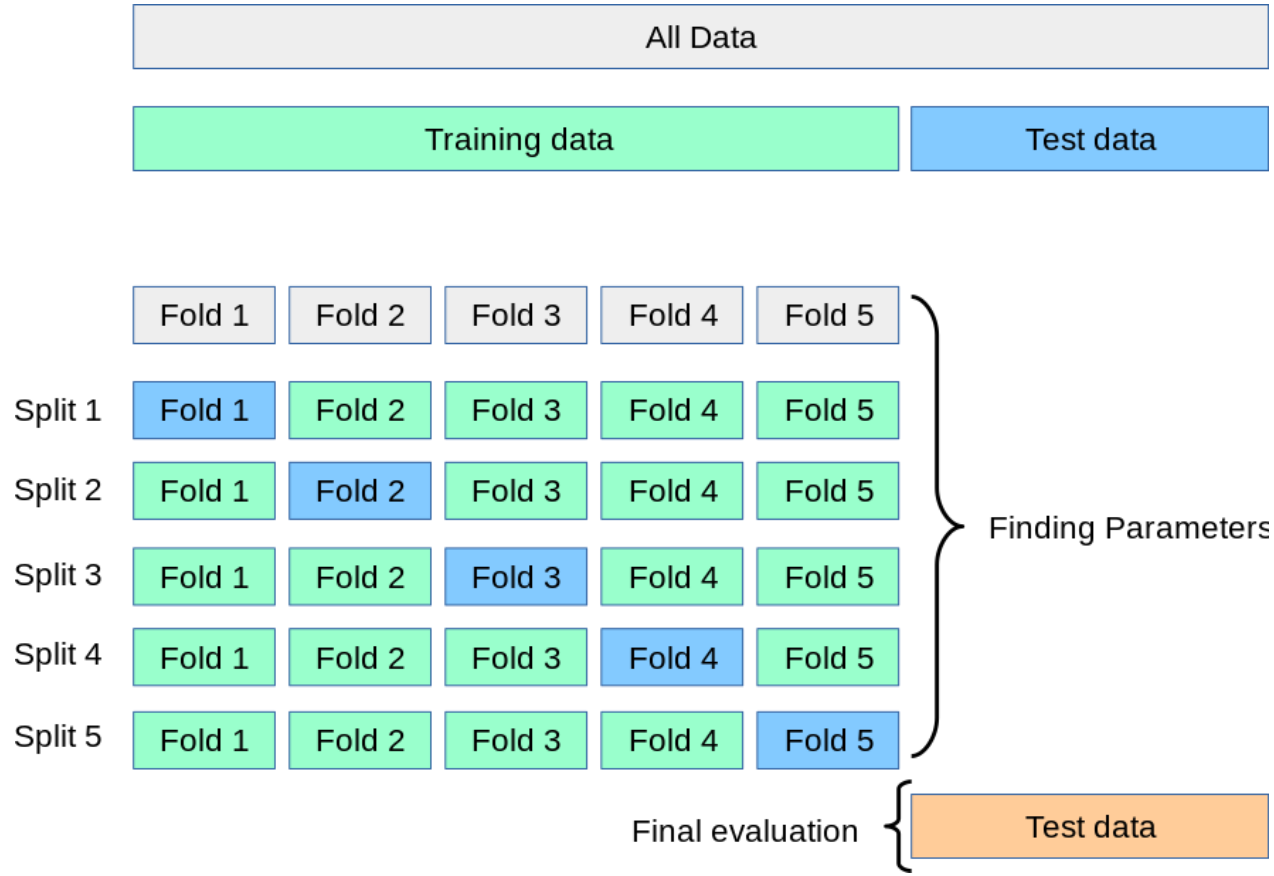
- Split data into train, validation, and test



K-fold Cross Validation

TEXAS A&M UNIVERSITY Engineering

- Train and validate model
  - K-fold CV (find best hyperparameters)
- Evaluate on hold out test data

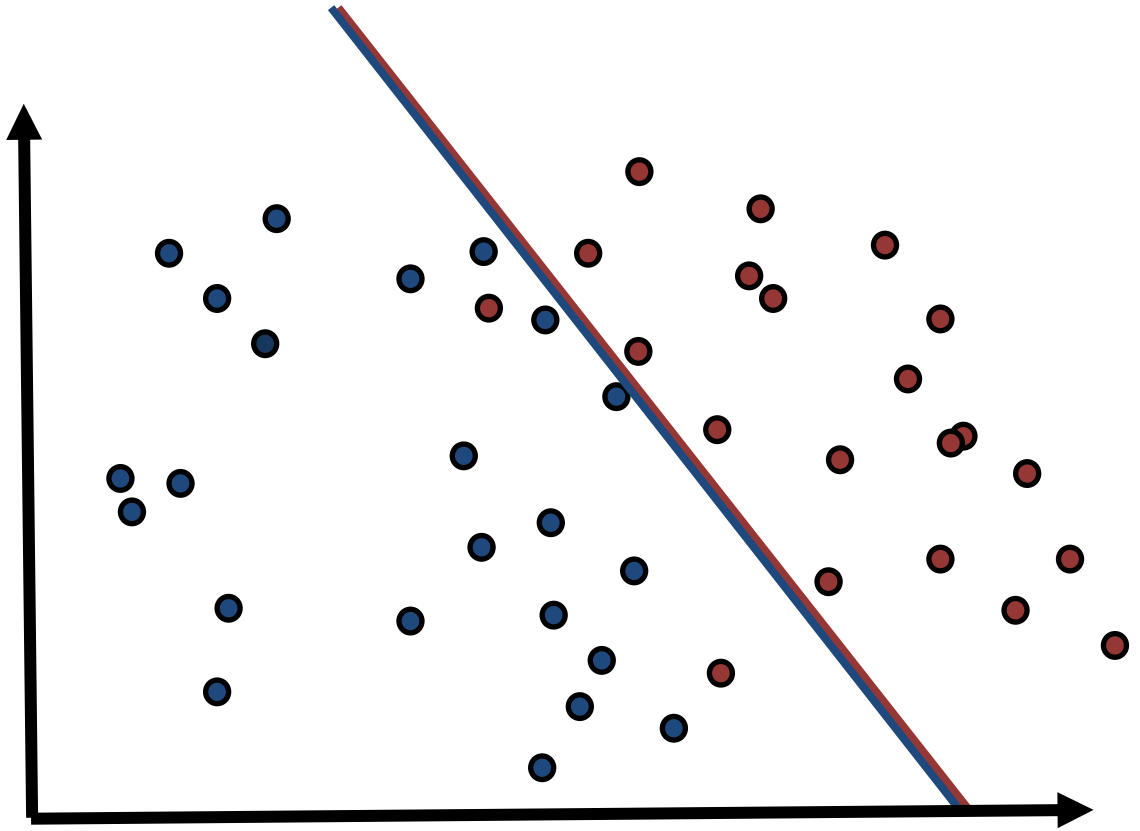


Classification Measures of Performance

Classification Performance Metrics

TEXAS A&M UNIVERSITY Engineering

- Evaluation (many metrics possible)
  - Accuracy: fraction of instances predicted correctly

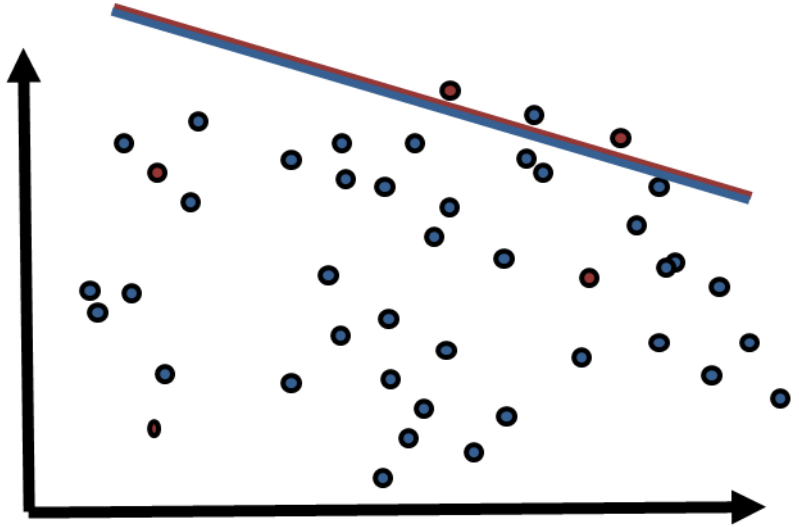


Blue samples = 23, Red samples = 21, Total samples = 44
Predicted incorrectly = 2
Accuracy = 42/44 = 95.45%

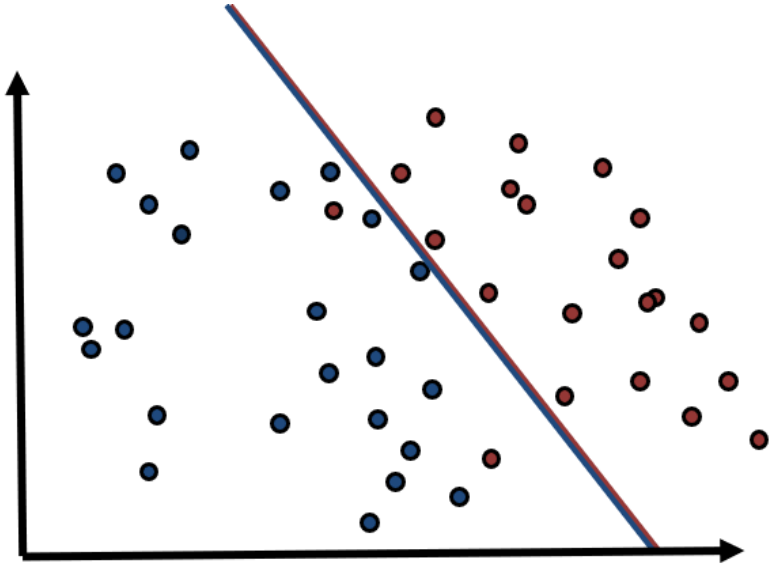
Classification Performance Metrics

TEXAS A&M UNIVERSITY Engineering

- Evaluation (many metrics possible)
  - Accuracy: fraction of instances predicted correctly
  - Not well suited for imbalanced classes



Blue samples = 39, Red samples = 5, Total samples = 44
Predicted incorrectly = 4
Accuracy = 42/44 = 90.90%

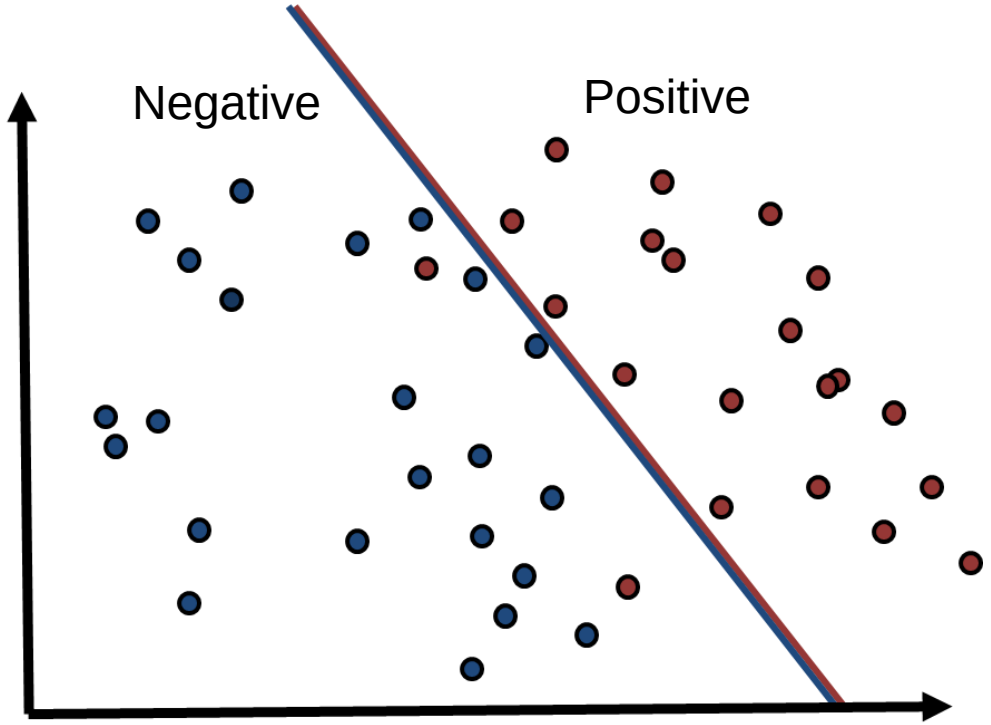


Blue samples = 23, Red samples = 21, Total samples = 44
Predicted incorrectly = 2
Accuracy = 42/44 = 95.45%

Errors in (Binary) Classification

TEXAS A&M UNIVERSITY Engineering

- Errors in classification:
  - True Positive – actual class = Positive; predicted class = Positive
  - True Negative – actual class = Negative; predicted class = Negative
  - False Positive – actual class = Negative; predicted class = Positive
  - False Negative – actual class = Positive; predicted class = Negative



Blue samples = 23, Red samples = 21, Total samples = 44
TP=19; TN = 23; FP = 0; FN = 2

Errors in (Binary) Classification

TEXAS A&M UNIVERSITY Engineering

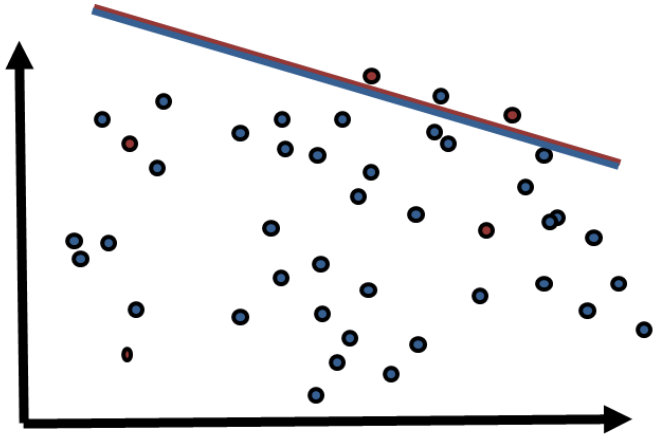
- Errors in classification:
  - True Positive – actual class = Positive; predicted class = Positive
  - True Negative – actual class = Negative; predicted class = Negative
  - False Positive – actual class = Negative; predicted class = Positive
  - False Negative – actual class = Positive; predicted class = Negative

	Predicted Negative	Predicted Positive
Actual Negative	True *Negative* ✓ Correctly predicted not dogs **3**	False Positive ✗ Predicted as dogs but actually not dogs **2**
Actual Positive	False *Negative* ✗ Predicted as not dogs but actually dogs **1**	True Positive ✓ Correctly predicted dogs **4**

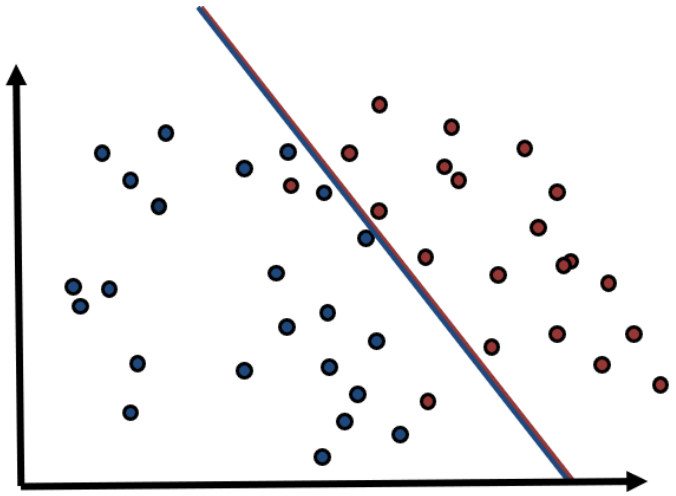
(Improved) Classification Performance Metrics

TEXAS A&M UNIVERSITY Engineering

- Evaluation (many metrics possible)
  - **Precision: TP / (TP + FP)**
  - **Recall: TP/ (TP + FN)**
  - **F_1 = Harmonic Mean of Precision & Recall**
  - **F_1 = TP / (TP + 0.5 FN + 0.5 FP)**

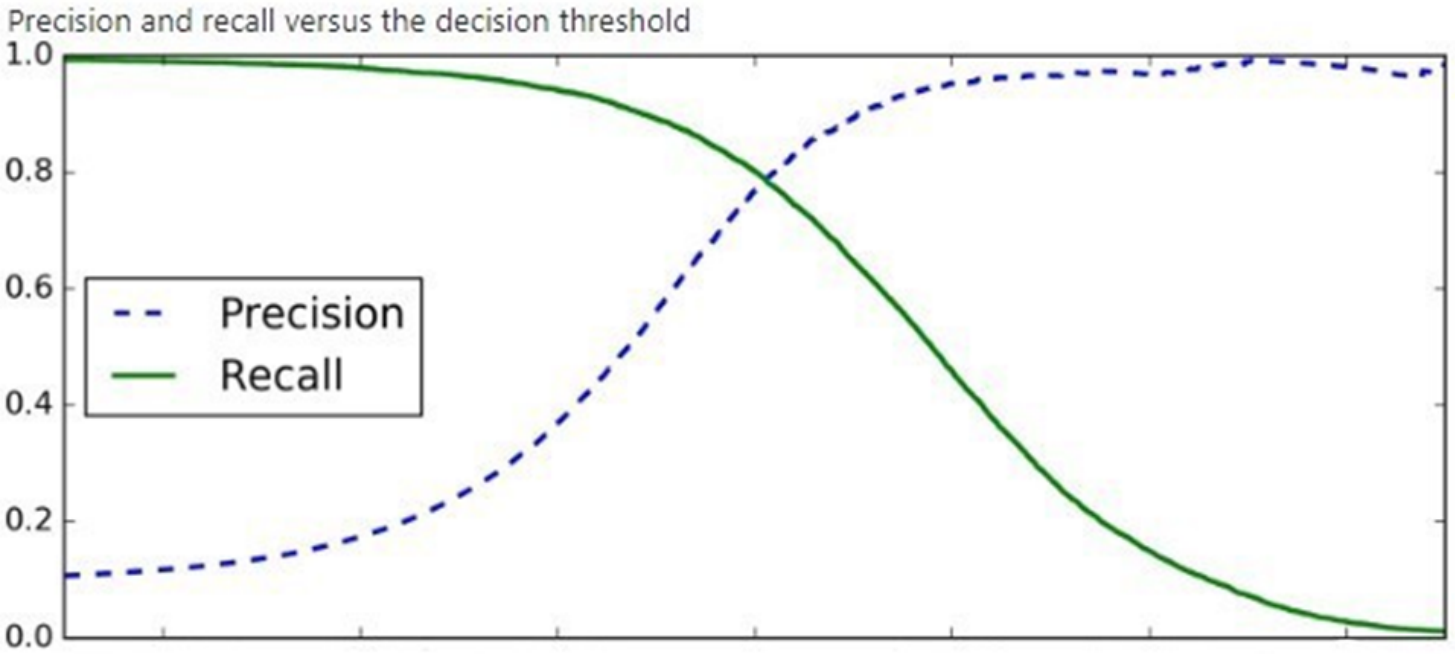


Blue samples = 39, Red Samples = 5, Total samples = 44
TP=2; TN=38; FP=1; FN=3
Accuracy = 40/44 = 90.9%
Precision = 2/3 = 66.7%
Recall = 2/5 = 40%
F_1=2/4 = 50%



Blue samples = 23, Red samples = 21, Total samples = 44
TP=19; TN=23; FP=0; FN=2
Accuracy = 42/44 = 95.5%
Precision = 100%
Recall = 19/21 = 90.5%
F_1=19/20 = 95%

Precision / Recall trade-off



Predict all positive

Only predict positive if certain positive

Precision: TP / (TP + FP)
Recall: TP/ (TP +FN)

Confusion Matrix

TEXAS A&M UNIVERSITY
Engineering

- Summarizes correct and incorrect classification by class
- For binary classifier:

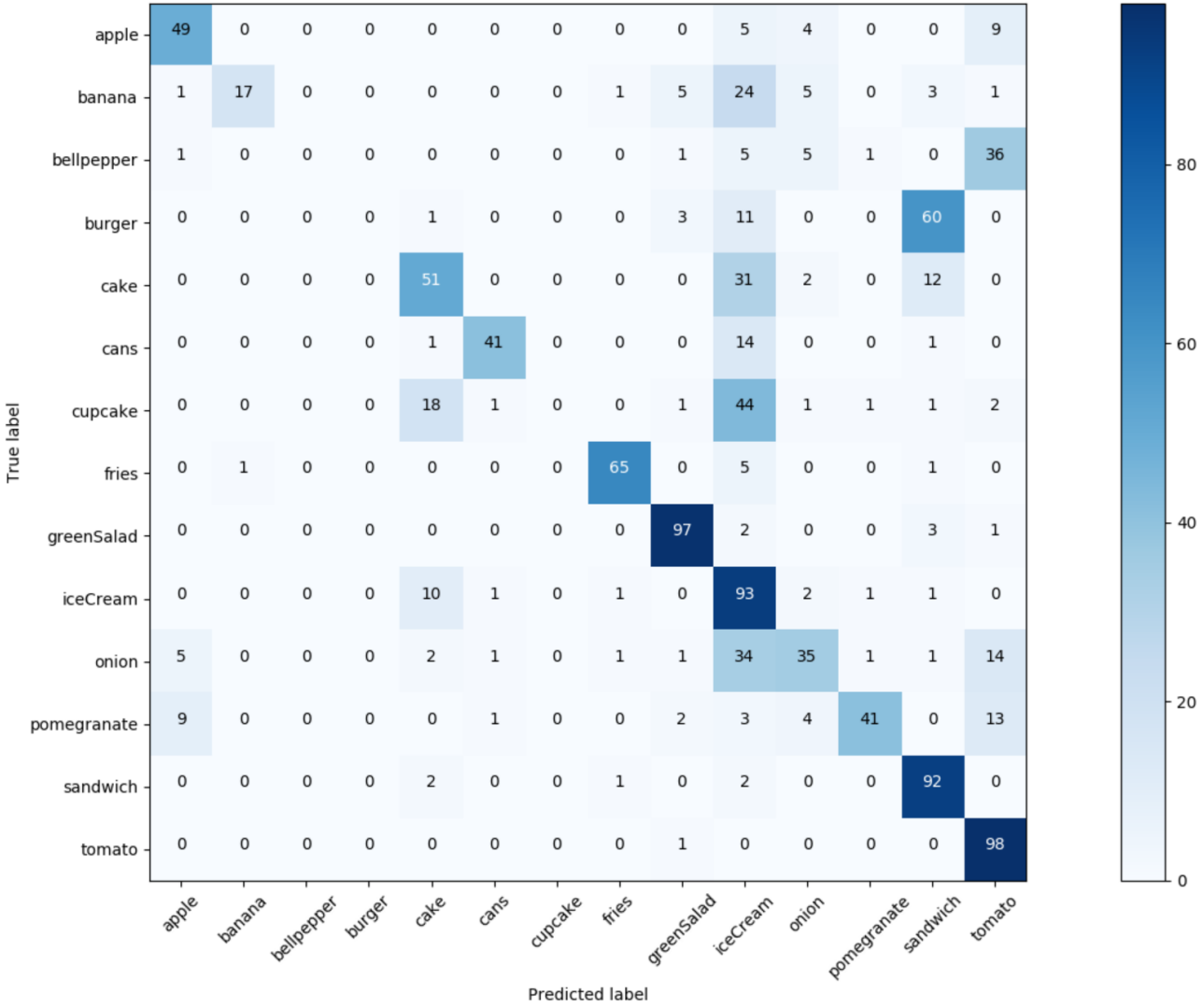
True Label	Positive	Negative
Positive	# True Positives	# False Negatives
Negative	# False Positives	# True Negatives

Predicted Label

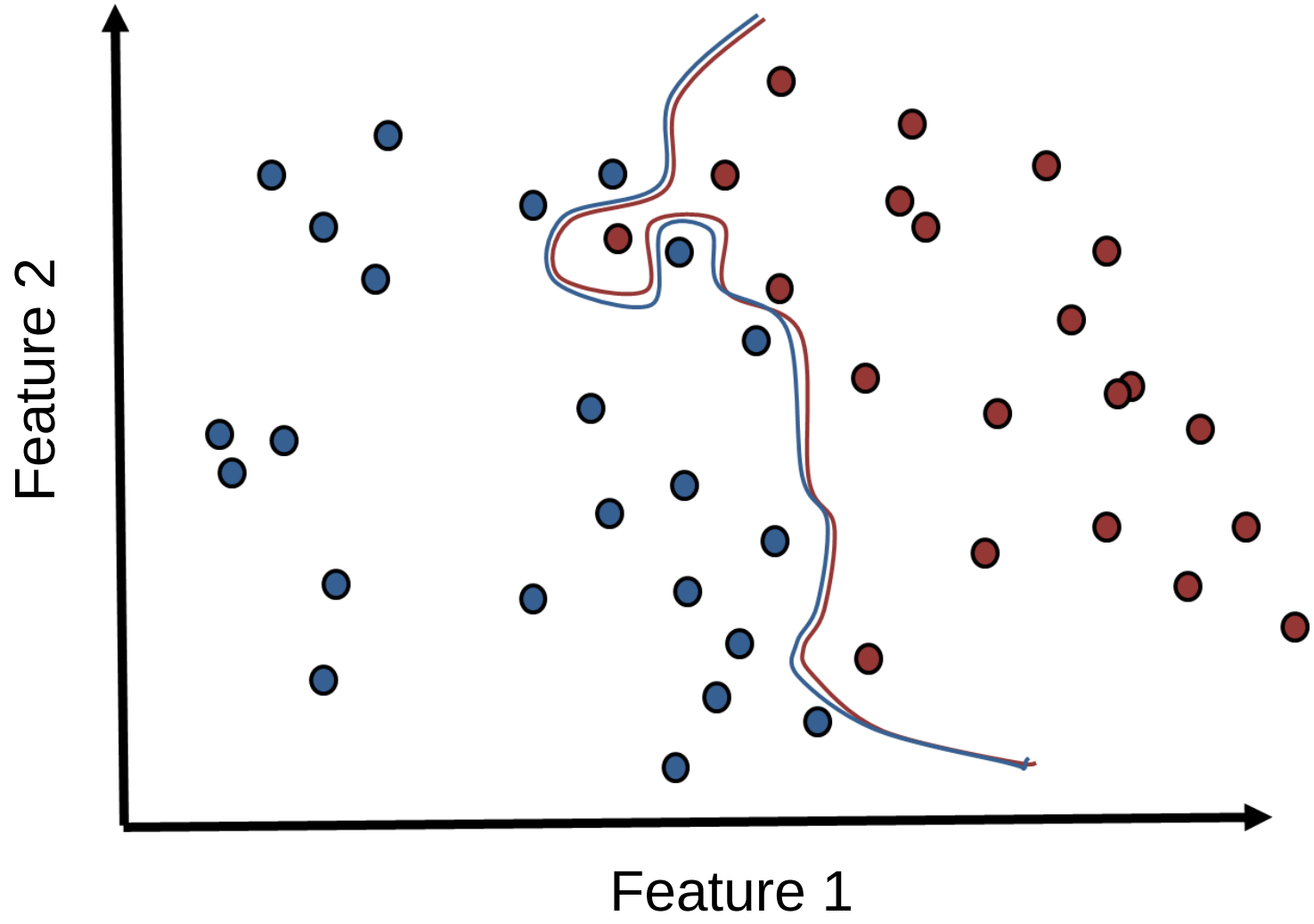
Multi-class Confusion Matrix

TEXAS A&M UNIVERSITY
Engineering

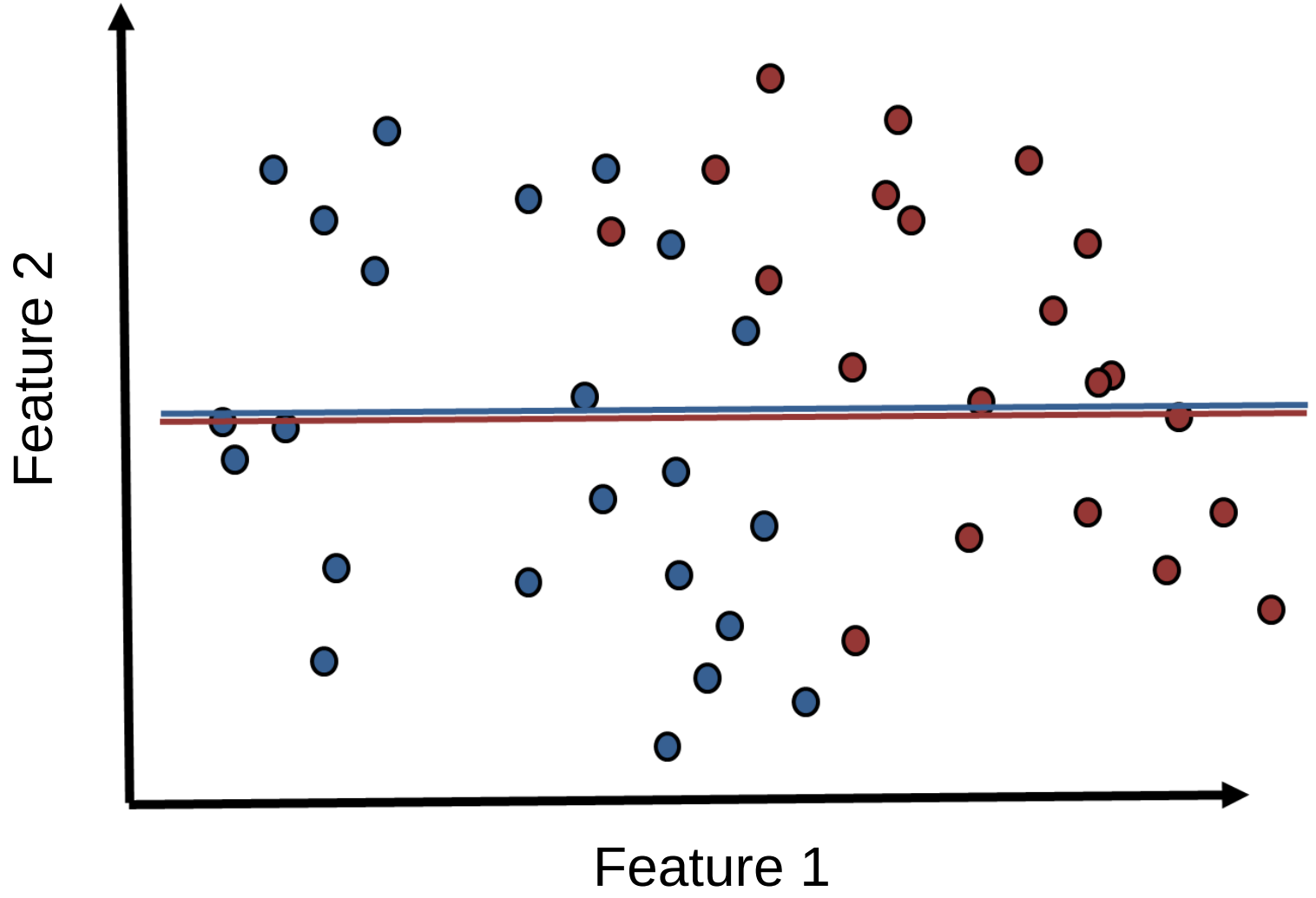
- For classification presents heat-map of classification results for ground truth classes versus predicted classes
- Very valuable in determining where classification overfitting is happening



Perfect Classifier?

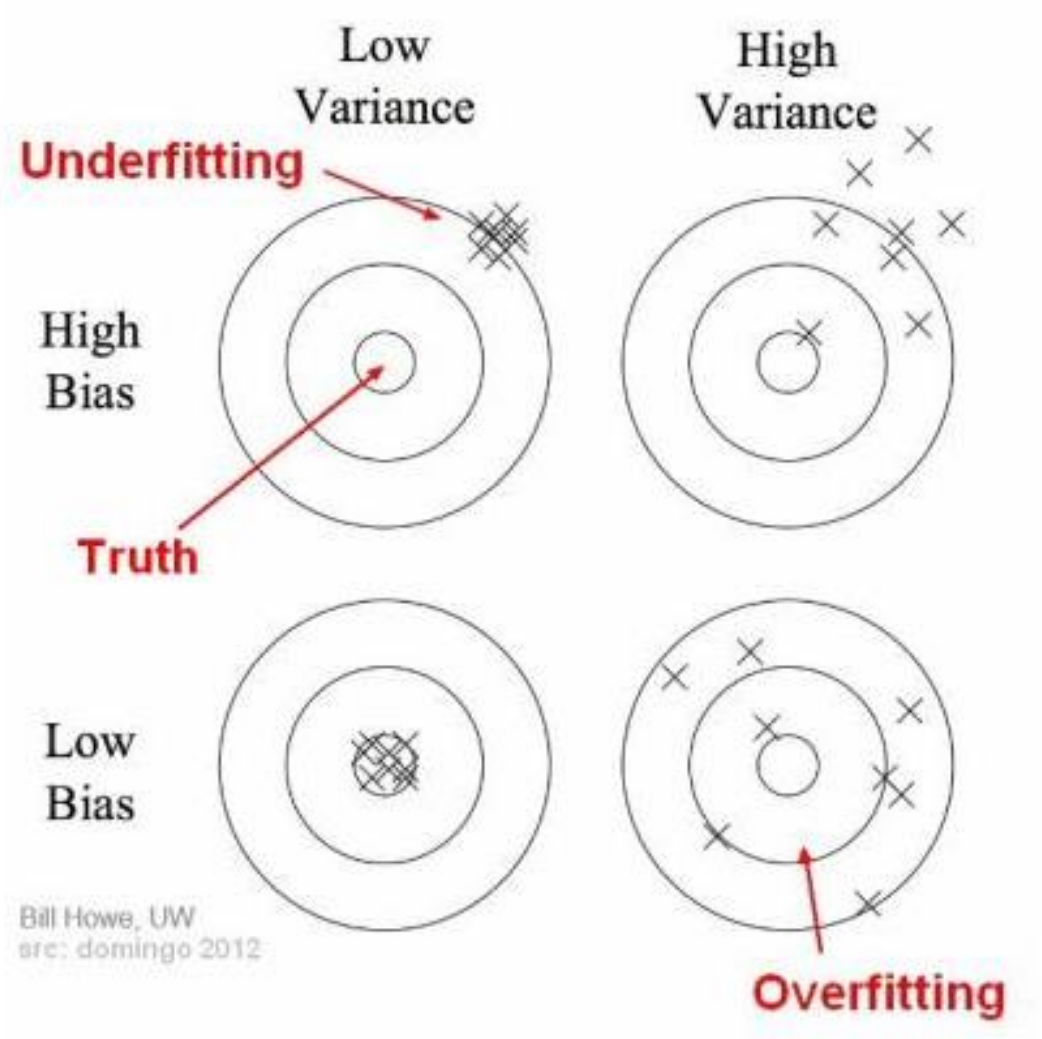


Equally bad? What's wrong?



Bias/Variance Tradeoff and Overfitting/Underfitting

Bias vs Variance



Underfitting and Overfitting

TEXAS A&M UNIVERSITY
Engineering

- **Underfitting:** If there is high Bias in the model we use (the model wrongly disregards the effects of the varying observations), the approximate function $\hat{Y}$ cannot closely approximate the unknown function Y
- **Overfitting:** If there is high Variance in the model we use (the model changes significantly with changes in the training data) the approximate function $\hat{Y}$ does not generalize well since the new observations differ from the training data)

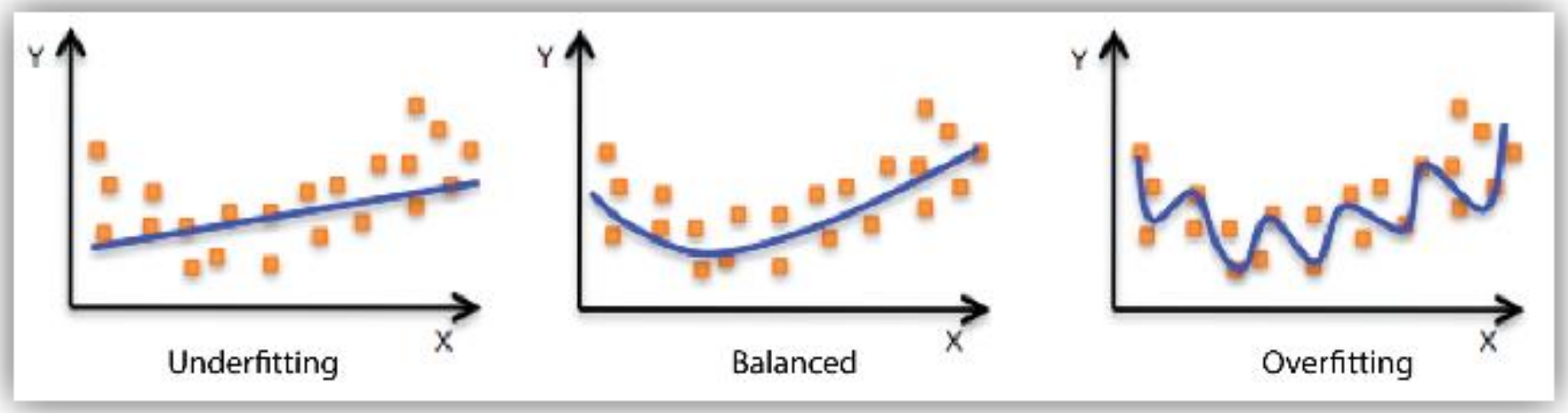


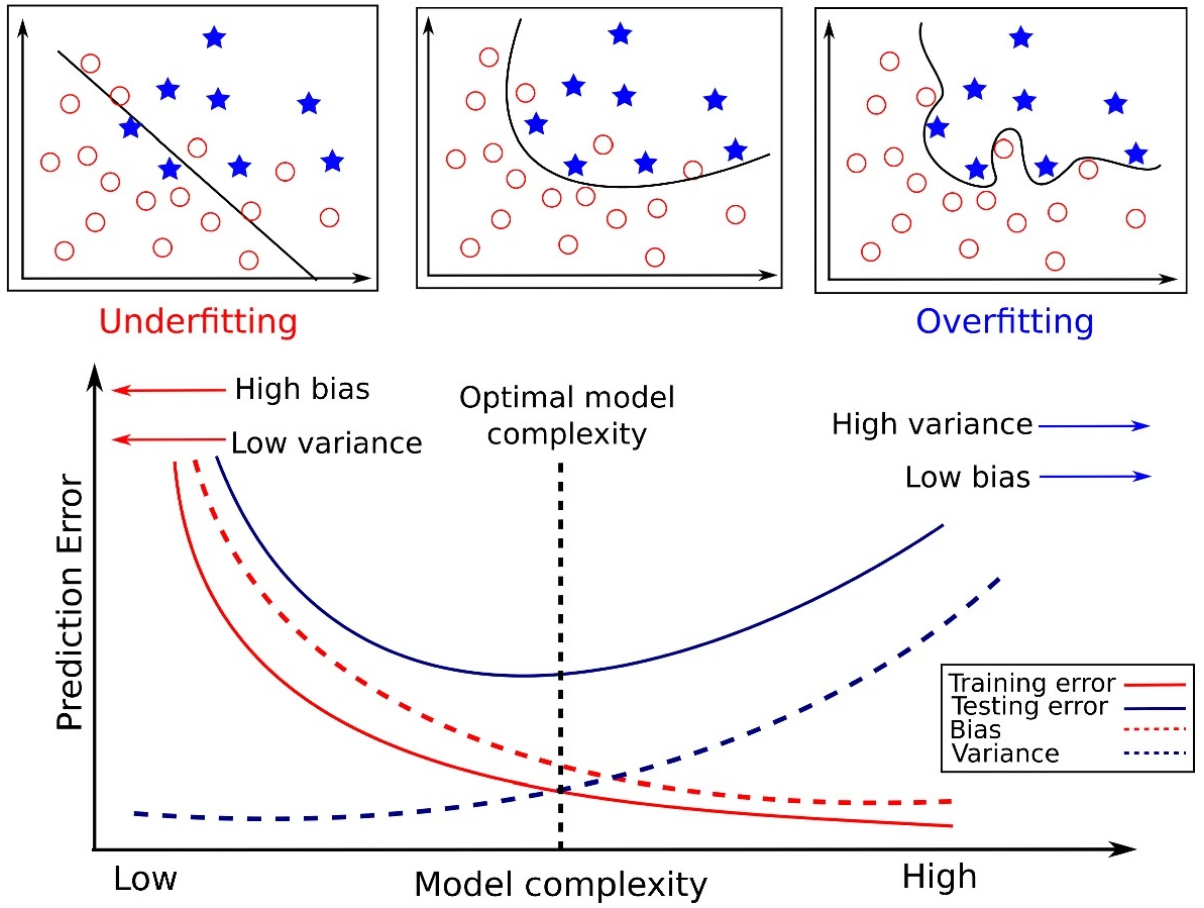
Image from: Analytics Vidhya, Regularization Technique in Linear Model

Decision Tree Classifier

Decision Trees

TEXAS A&M UNIVERSITY
Engineering

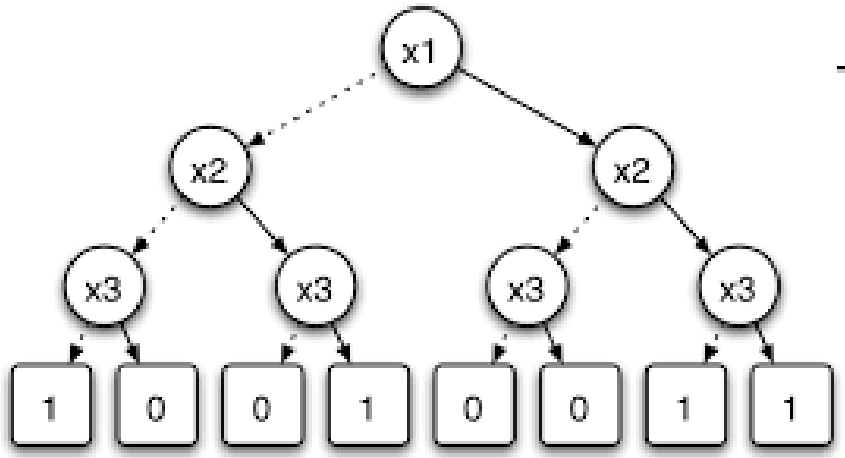
- Classification built on induction
- Suited to managing simplicity vs consistency (bias vs variance)



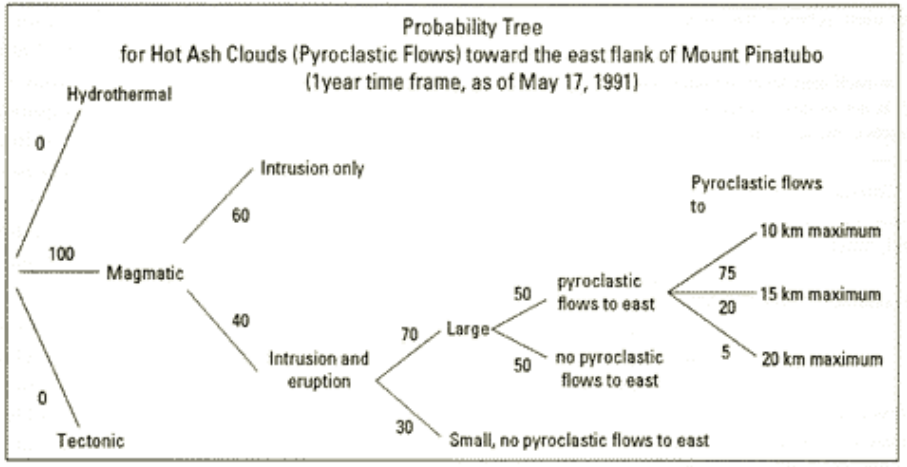
Decision Trees

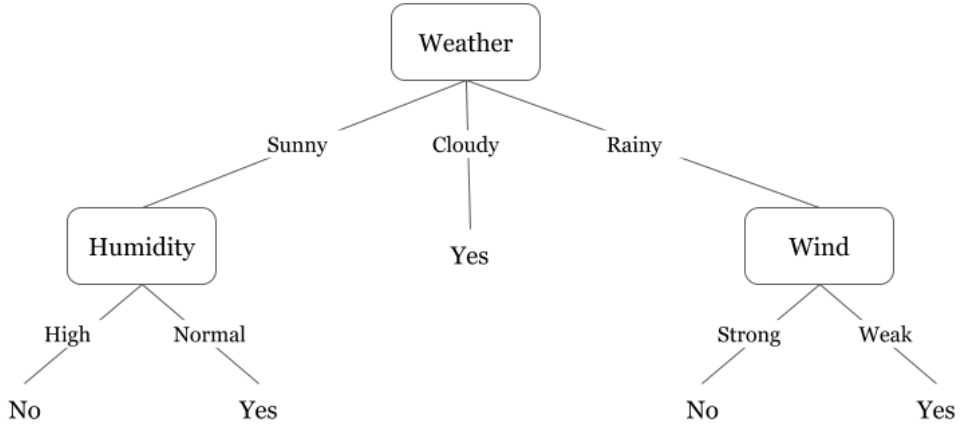
TEXAS A&M UNIVERSITY
Engineering

- Compact representation of a function:
  - Truth table
  - Conditional probability table
  - Predictions



x1	x2	x3	f
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1





Decision Trees to Rules

TEXAS A&M UNIVERSITY Engineering

$\mathcal{R}_3$: If $X_1 \leq 5.45$ and $X_2 \leq 2.8$ and $X_1 \leq 4.7$, then class is c_1, or

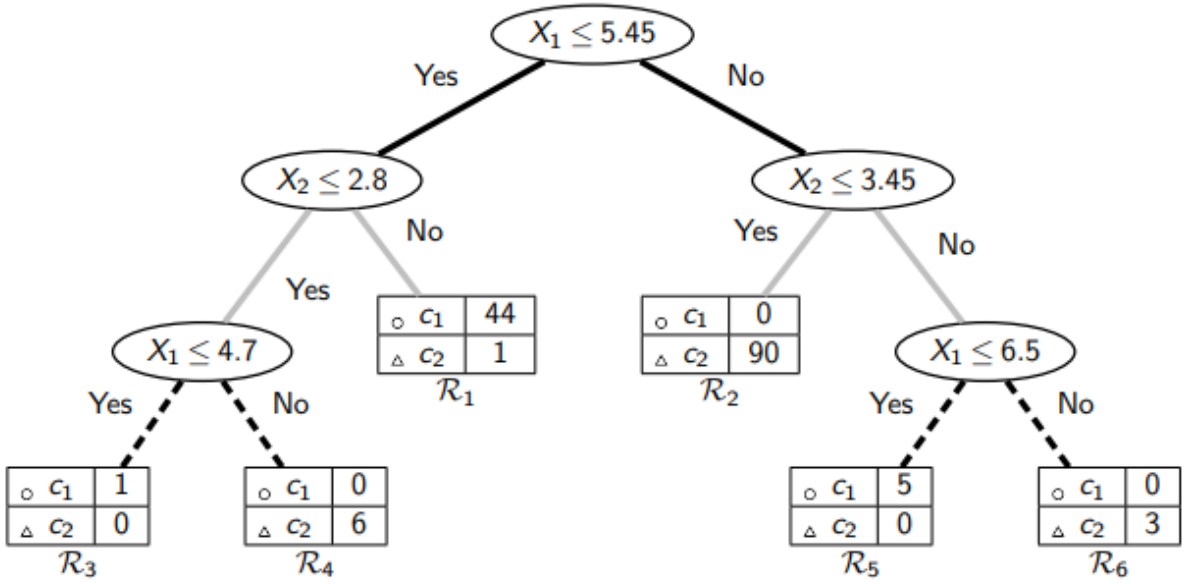
$\mathcal{R}_4$: If $X_1 \leq 5.45$ and $X_2 \leq 2.8$ and $X_1 > 4.7$, then class is c_2, or

$\mathcal{R}_1$: If $X_1 \leq 5.45$ and $X_2 > 2.8$, then class is c_1, or

$\mathcal{R}_2$: If $X_1 > 5.45$ and $X_2 \leq 3.45$, then class is c_2, or

$\mathcal{R}_5$: If $X_1 > 5.45$ and $X_2 > 3.45$ and $X_1 \leq 6.5$, then class is c_1, or

$\mathcal{R}_6$: If $X_1 > 5.45$ and $X_2 > 3.45$ and $X_1 > 6.5$, then class is c_2



Decision Tree Learning Algorithm

Example Dataset

TEXAS A&M UNIVERSITY Engineering

2004 model year vehicle data and classes of fuel economy

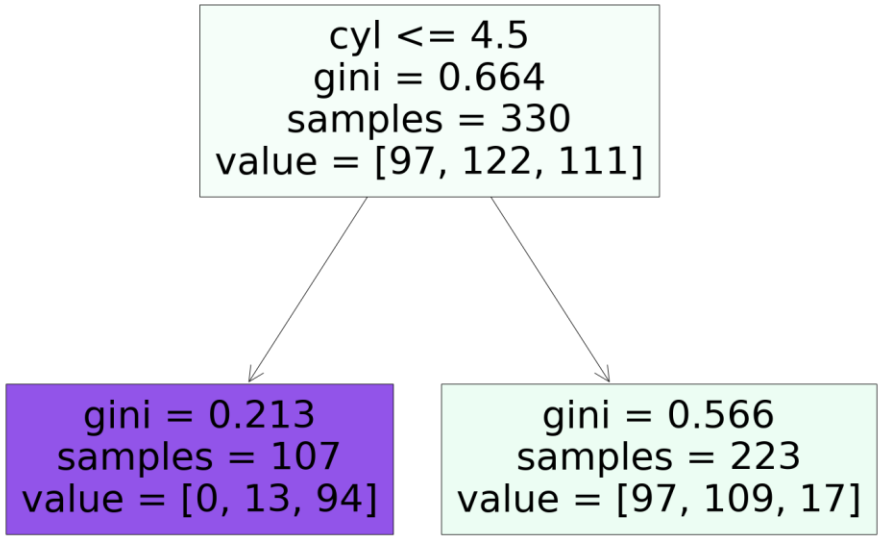
```
df.head(10)
```

	msrp	invoice	disp	cyl	hp	weight	wheelbase	length	width	AWD	RWD	mpgclass
0	22388	20701	1.8	4.0	142	2387.0	89.0	156.0	66.0	0	1	2.0
1	35545	32244	3.8	6.0	205	3778.0	114.0	207.0	75.0	0	0	1.0
2	39235	36052	4.0	8.0	270	3953.0	106.0	185.0	69.0	0	0	1.0
3	45707	41966	3.2	6.0	215	3770.0	107.0	183.0	69.0	0	1	1.0
5	34845	32902	2.3	5.0	247	3766.0	107.0	180.0	71.0	0	0	1.0
6	35515	32243	3.2	6.0	220	5086.0	112.0	187.0	76.0	1	0	0.0
7	14610	13697	2.2	4.0	140	2617.0	104.0	183.0	69.0	0	0	2.0
8	33112	30763	3.8	6.0	215	4718.0	110.0	190.0	75.0	1	0	0.0
9	21900	20095	3.4	6.0	180	3465.0	111.0	200.0	73.0	0	0	2.0
11	25995	23969	2.5	6.0	174	3577.0	101.0	175.0	71.0	1	0	1.0

Decision Tree Learning

TEXAS A&M UNIVERSITY Engineering

- Training: Find a small tree consistent with the training examples (think Occam's razor), by:
- Recursively choosing the "most significant" attribute for each node which is eligible for expansion



We need a measure of how "good" a split is, even if the results aren't perfectly separated

Gini Impurity

- A decision in which all members are of one class is “pure”
- Decisions with mixed classifications are “impure”
- In decision-tree classification, the criteria splits are made using the Gini index

$$G(\boldsymbol{D}) = 1 - \sum_{i=1}^{k} P(c_i|\boldsymbol{D})^2$$

Gini Impurity

- A decision in which all members are of one class is “pure”
- Decisions with mixed classifications are “impure”
- In decision-tree classification, the criteria splits are made using the Gini index

Concretely, for a set of items with K classes, and p_k being the fraction of items labeled with class $k \in 1, 2, \ldots, K$, the **Gini impurity** is defined as:

$$G = \sum_{k=1}^{K} p_k(1 - p_k) = 1 - \sum_{k=1}^{N} p_k^2$$

Entropy

- Alternatively we can make decisions on which features to use in a split by using Entropy
- Measure of disorder or uncertainty
  - Low entropy if pure (entropy = 0)
  - Higher entropy if unpure (entropy = $\log_2 k$)

$$H(\boldsymbol{D}) = -\sum_{i=1}^{k} P(c_i|\boldsymbol{D}) \log_2 P(c_i|\boldsymbol{D})$$

Information Gain

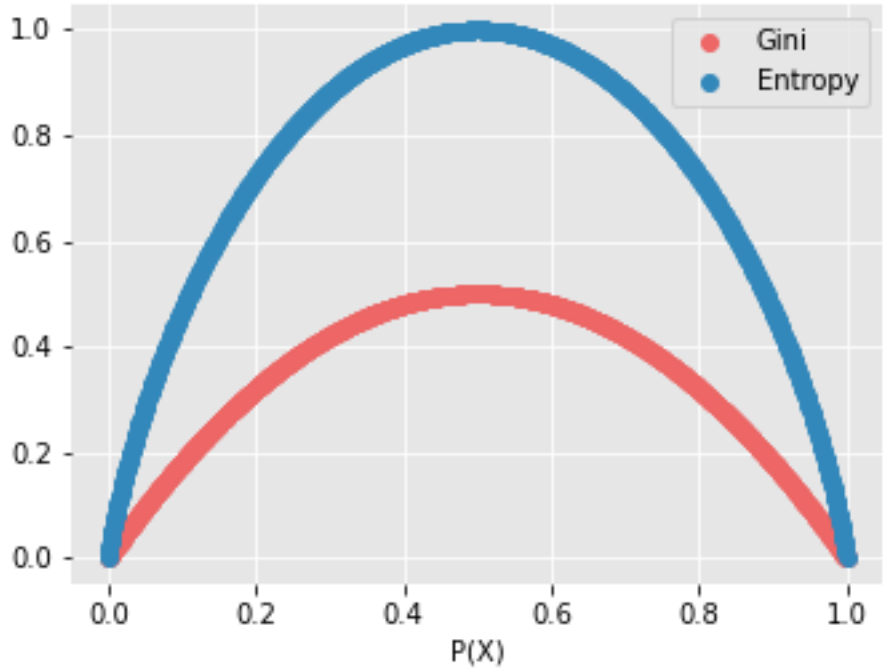
- For each split, compare entropy before and after
  - Difference is the information gain
- Problem: there's more than one distribution after split!
- Solution: use expected entropy, weighted by the number of examples

$$H(\boldsymbol{D}_Y, \boldsymbol{D}_N) = \frac{n_Y}{n} H(\boldsymbol{D}_Y) + \frac{n_N}{n} H(\boldsymbol{D}_N)$$

$$Gain(\boldsymbol{D}, \boldsymbol{D}_Y, \boldsymbol{D}_N) = H(\boldsymbol{D}) - H(\boldsymbol{D}_Y, \boldsymbol{D}_N)$$

Information gain = entropy (parent) − [weightes average] * entropy (children)

Gini vs Entropy



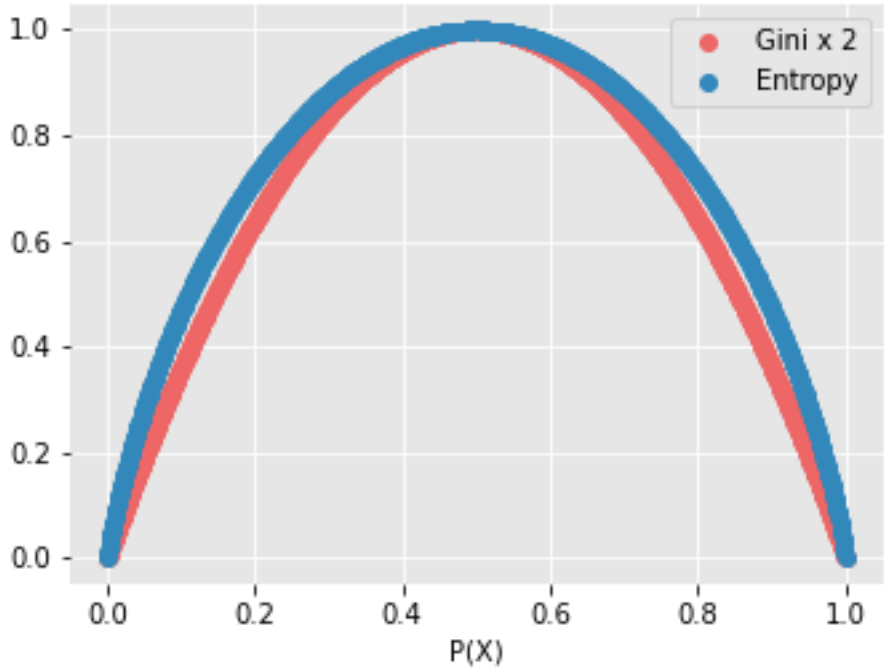


Image from: Pablo Aznar, Decision Trees: Gini vs Entropy.

Decision Tree Terminology

TEXAS A&M UNIVERSITY
Engineering

- Parent node: Node that is subdivided
- Child node: Sub-nodes of parents

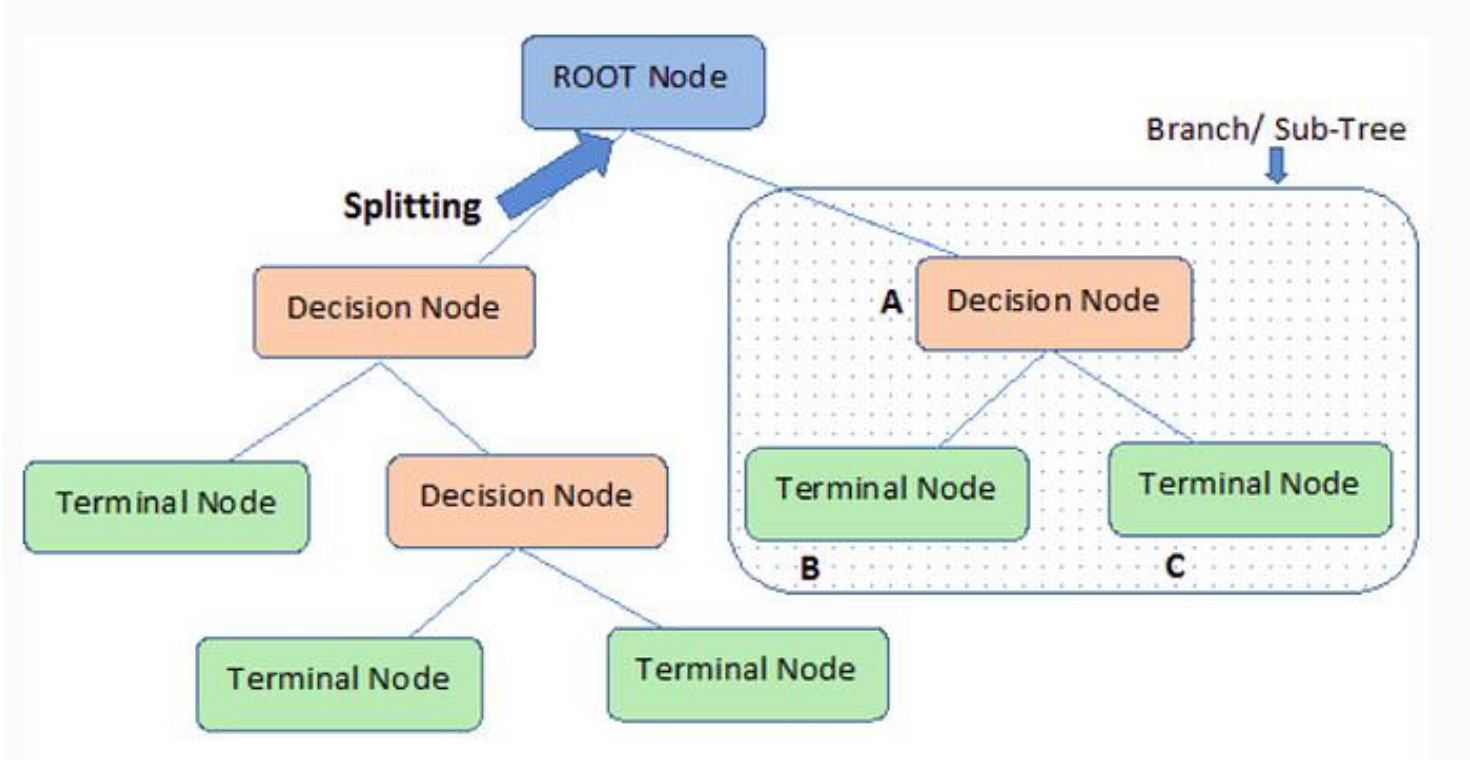
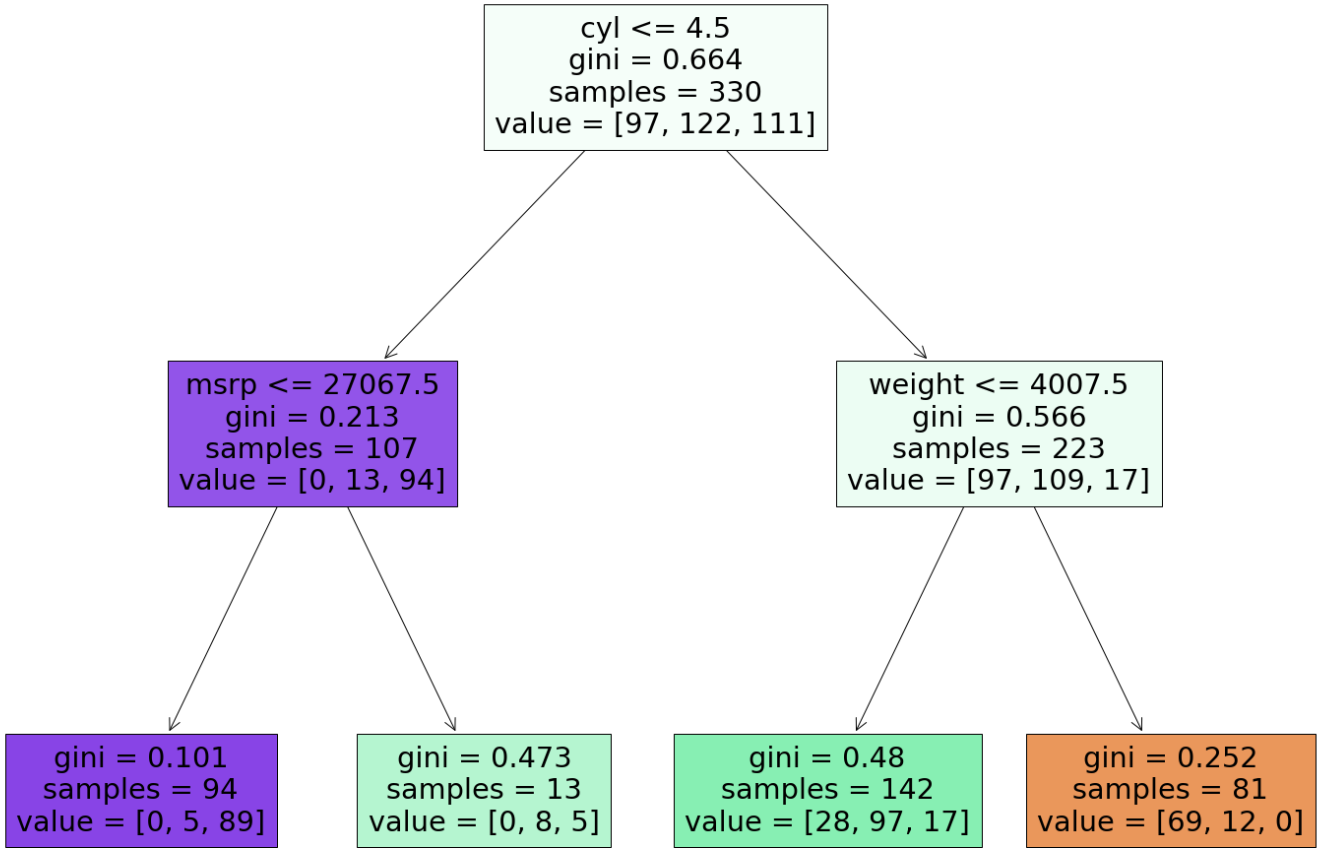


Image from: Arif R, The Basics of Decision Trees.

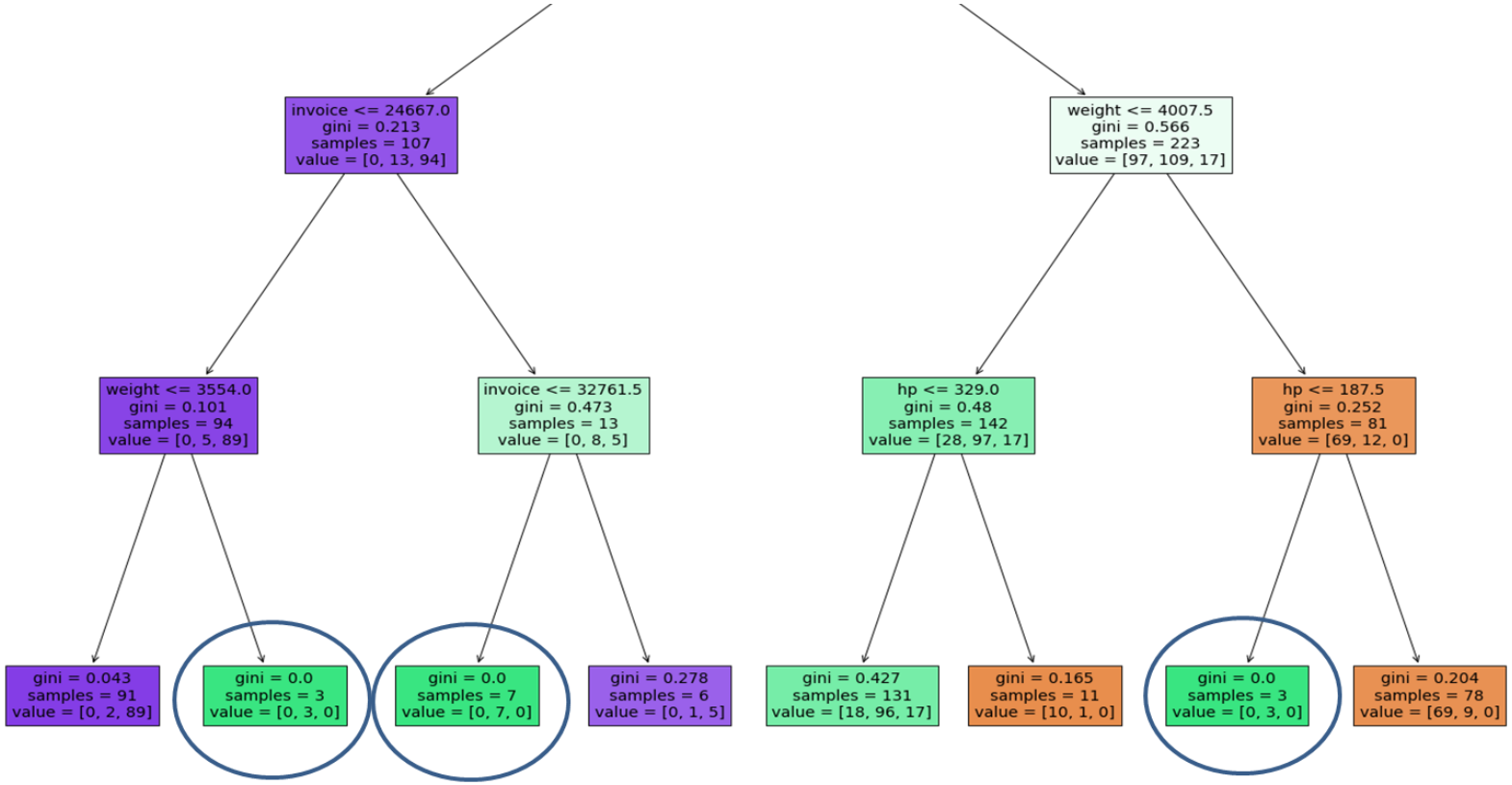
Decision Tree: Recursion



Decision Tree: Recursion

TEXAS A&M UNIVERSITY
Engineering

- Stop when states are “pure” or maximum depth is reached



Random Forest Classifier

Overfitting

TEXAS A&M UNIVERSITY
Engineering

- Overfitting occurs when you move from modeling the underlying patterns in the true function to also modeling the noise in the data
- Detecting when we overfit
  - The error in our **validation** or test set is significantly worse than our training set

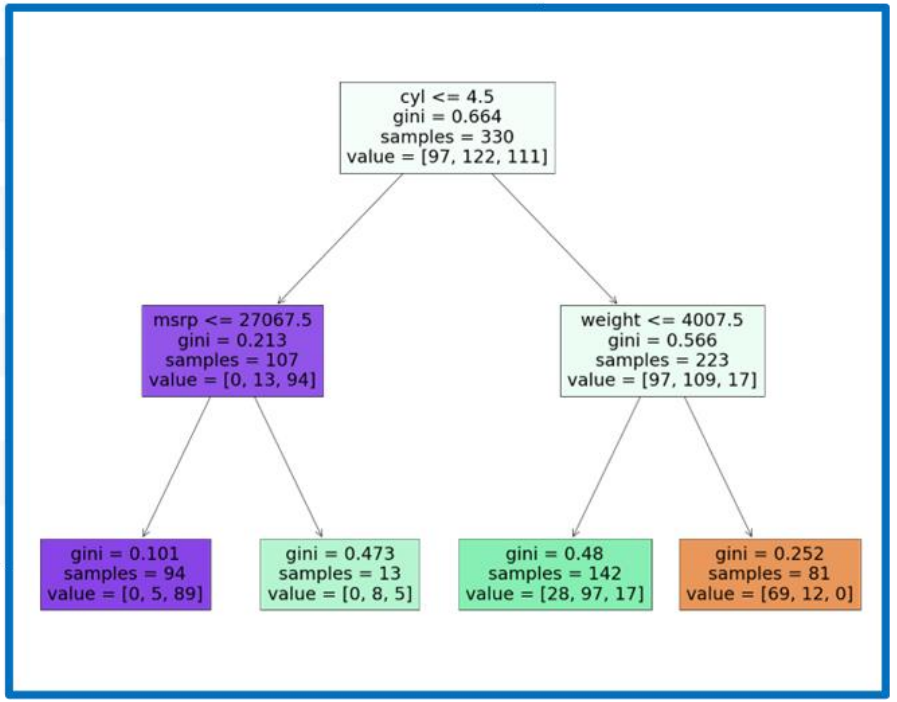
Recall this Decision Tree

TEXAS A&M UNIVERSITY
Engineering

2004 model year vehicle data and classes of fuel economy

```
df.head(10)
```

	msrp	invoice	disp	cyl	hp	weight	wheelbase	length	width	AWD	RWD	mpgclass
0	22388	20701	1.8	4.0	142	2387.0	89.0	156.0	66.0	0	1	2.0
1	35545	32244	3.8	6.0	205	3778.0	114.0					
2	39235	36052	4.0	8.0	270	3953.0	106.0					
3	45707	41966	3.2	6.0	215	3770.0	107.0					
5	34845	32902	2.3	5.0	247	3766.0	107.0					
6	35515	32243	3.2	6.0	220	5086.0	112.0					
7	14610	13697	2.2	4.0	140	2617.0	104.0					
8	33112	30763	3.8	6.0	215	4718.0	110.0					
9	21900	20095	3.4	6.0	180	3465.0	111.0					
11	25995	23969	2.5	6.0	174	3577.0	101.0					



Checking for Overfitting

TEXAS A&M UNIVERSITY Engineering

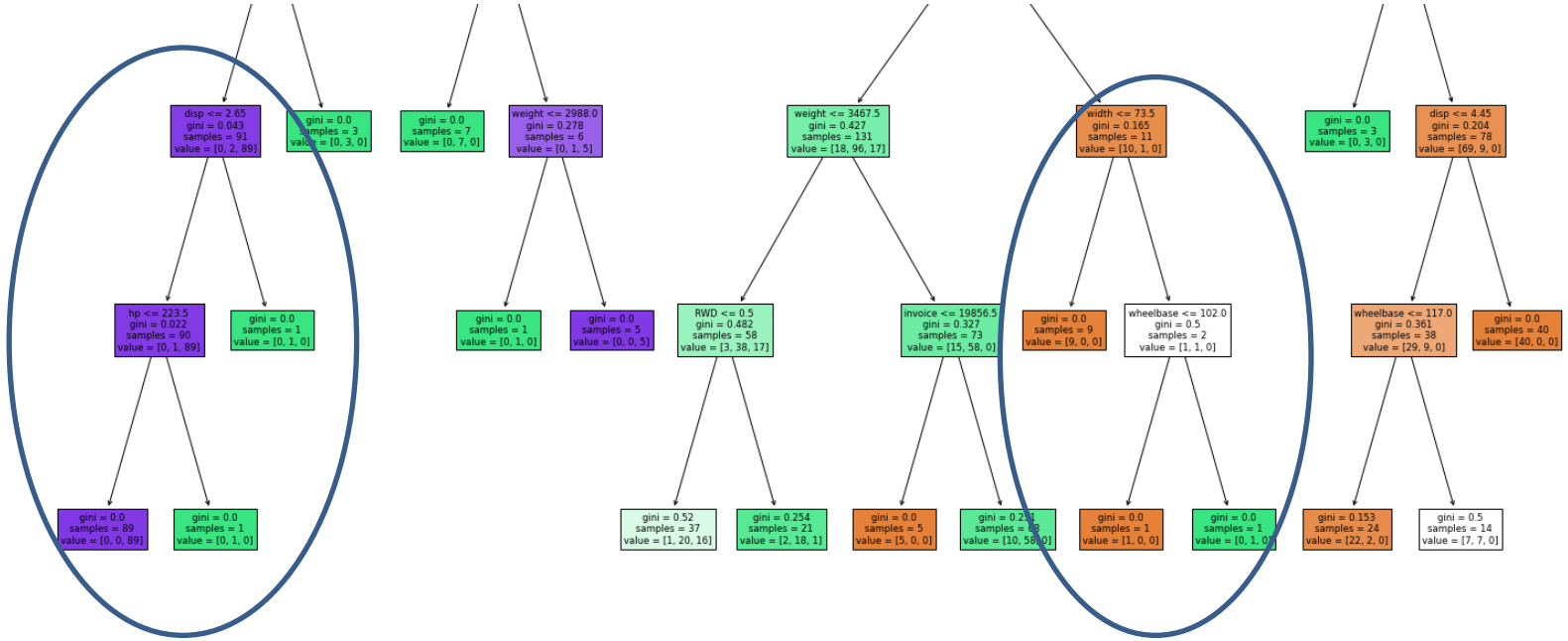
- When we build decision trees allowing different depths, we get a variety of model complexity. For this dataset, we get this accuracy for these models:

Max Tree Depth	Train Accuracy	Validation Accuracy
1	0.616	0.711
2	0.797	0.735
3	0.855	0.735
4	0.861	0.735
5	0.882	0.759
6	0.918	0.759
13	0.918	0.759

Mitigating Overfitting

TEXAS A&M UNIVERSITY
Engineering

- **Tree Pruning**
- Regularization



Mitigating Overfitting

TEXAS A&M UNIVERSITY
Engineering

- Tree Pruning
- **Regularization**

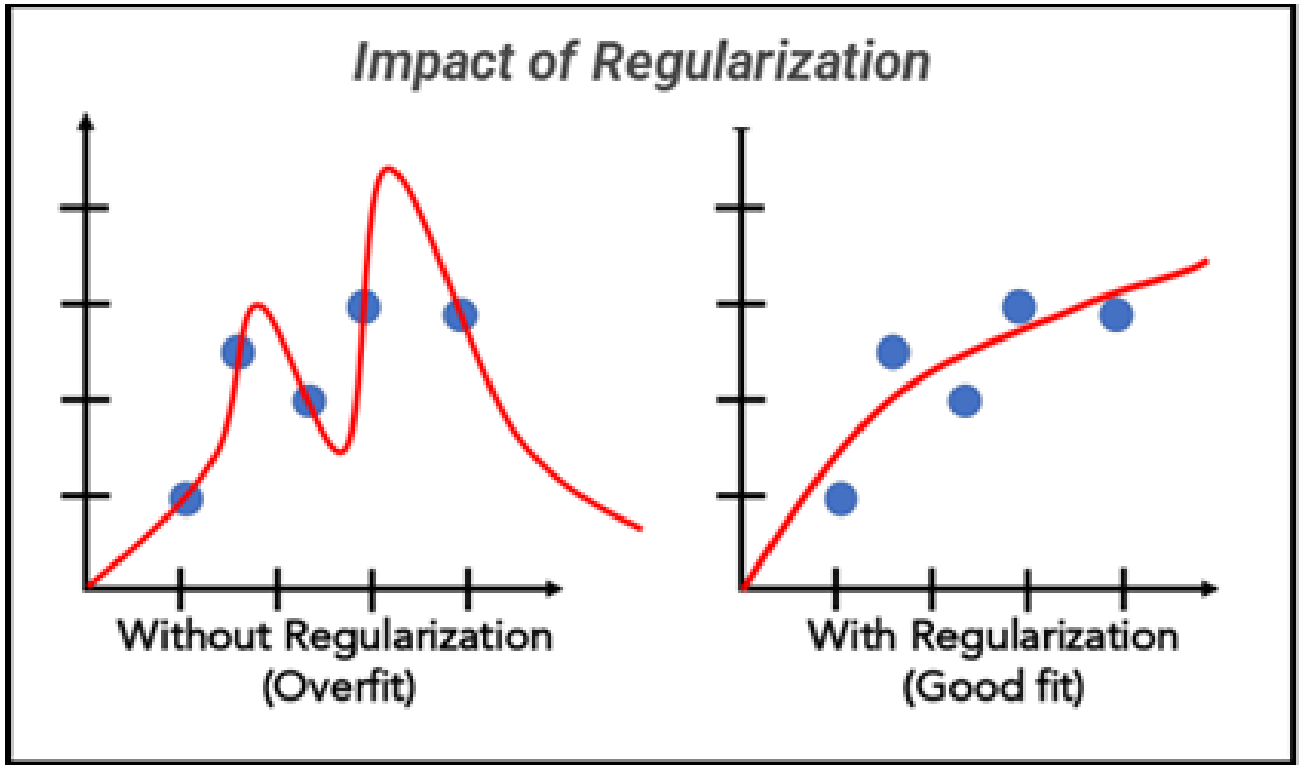


Image from: Analytics Vidhya, Regularization Technique in Linear Model

Random Forests

TEXAS A&M UNIVERSITY
Engineering

- Aggregate multiple decisions trees to form the predictors

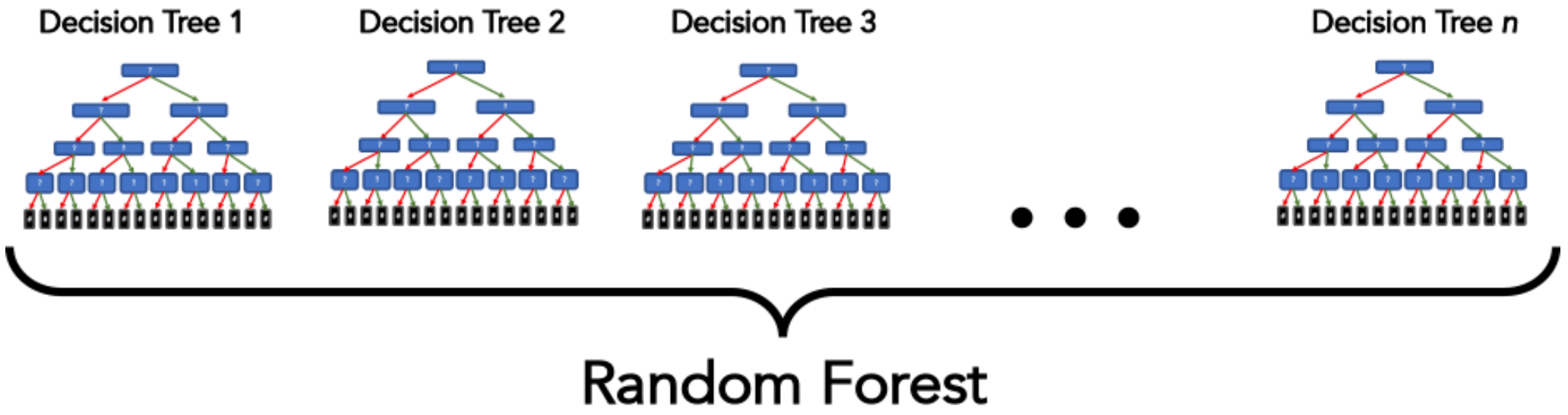


Image from: Thaddeus Segura, Random Forest Regression in 200 words.

Ensemble Learning

Ensemble Learning

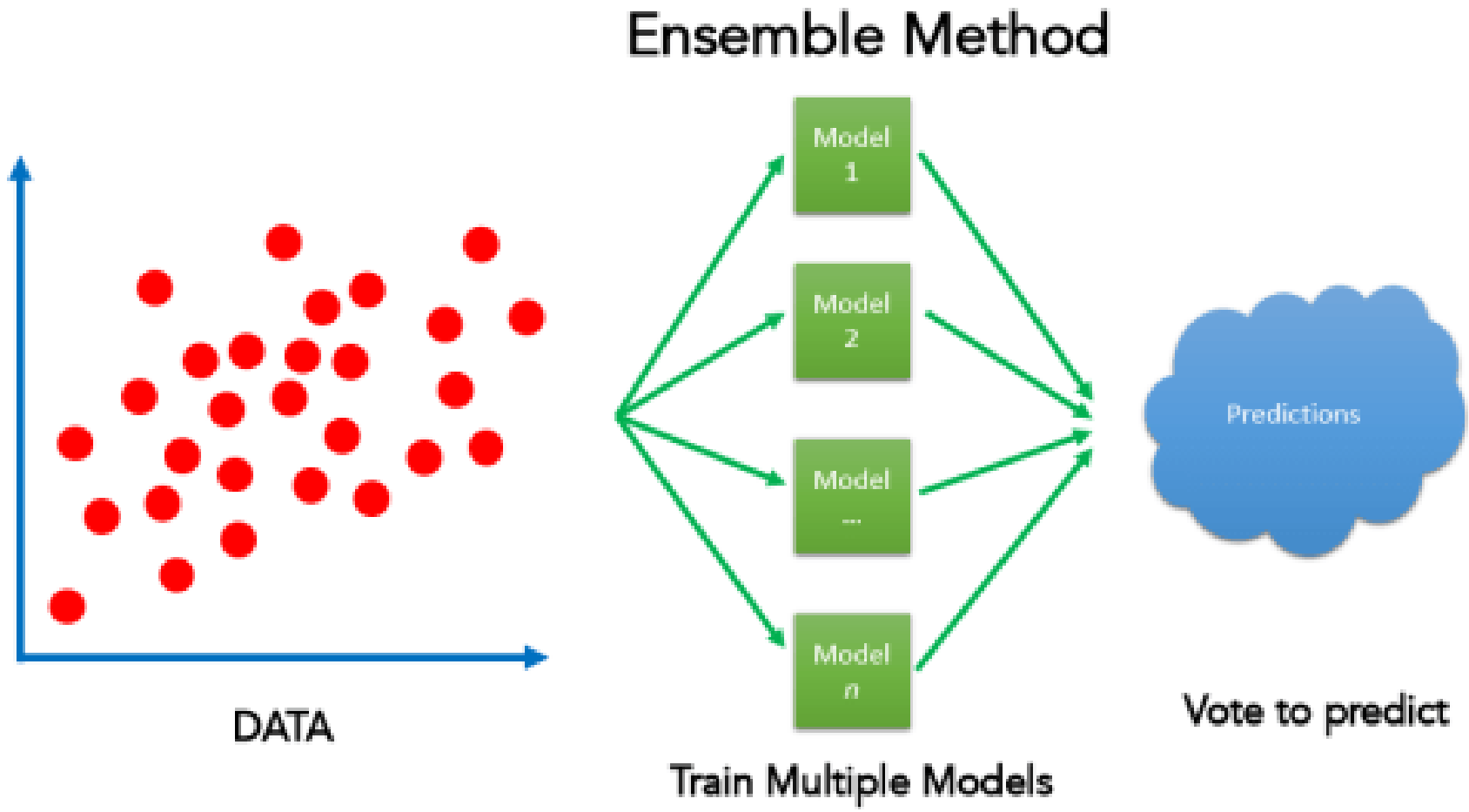


Image from: Thaddeus Segura, Random Forest Regression in 200 words.

Ensemble Learning

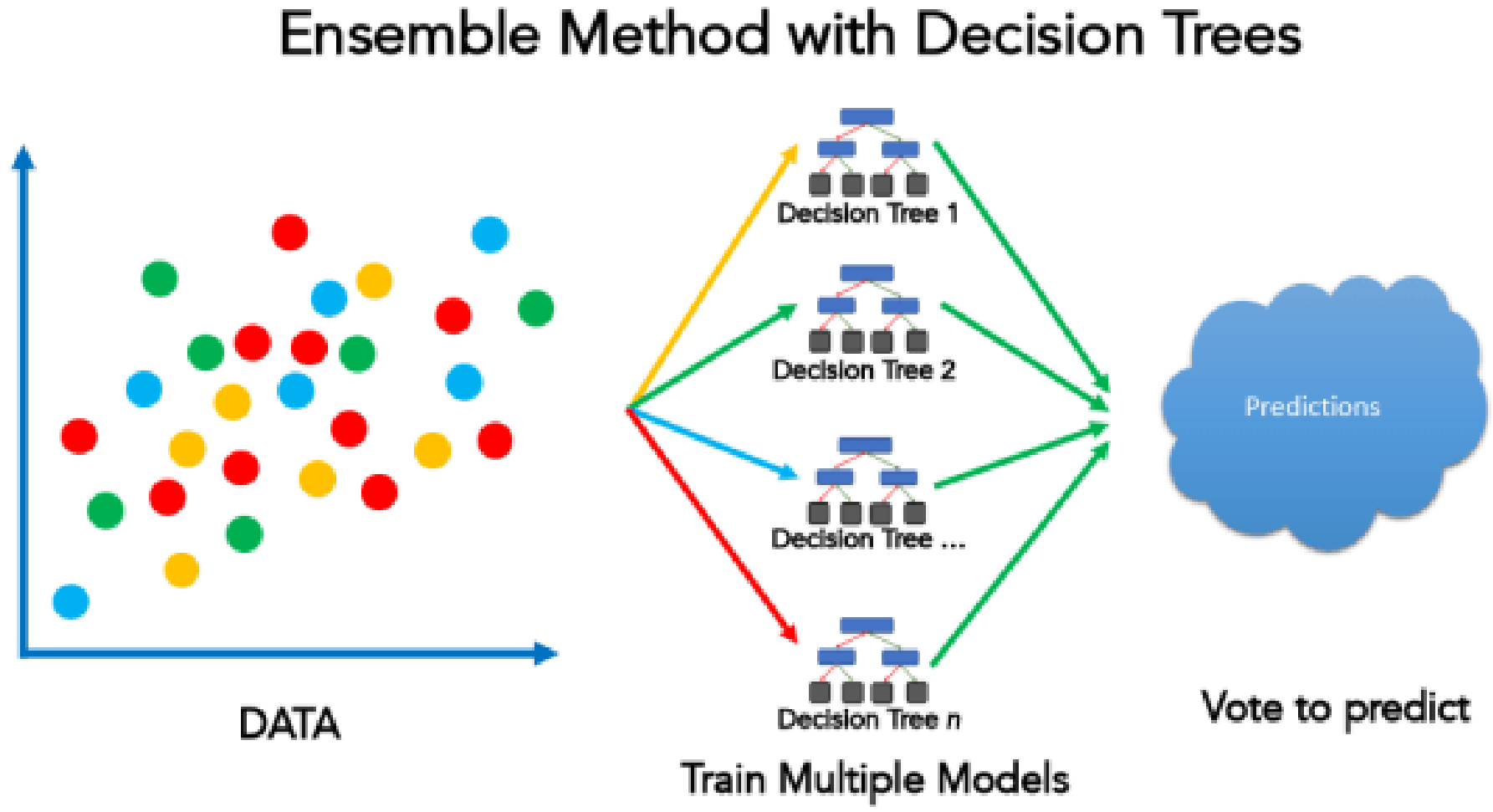


Image from: Thaddeus Segura, Random Forest Regression in 200 words.

Ensemble Learning

TEXAS A&M UNIVERSITY
Engineering

- Aggregate groups of predictors
  - Train classifiers and majority vote
  - Improves overall accuracy, even if component classifiers are “weak learners”

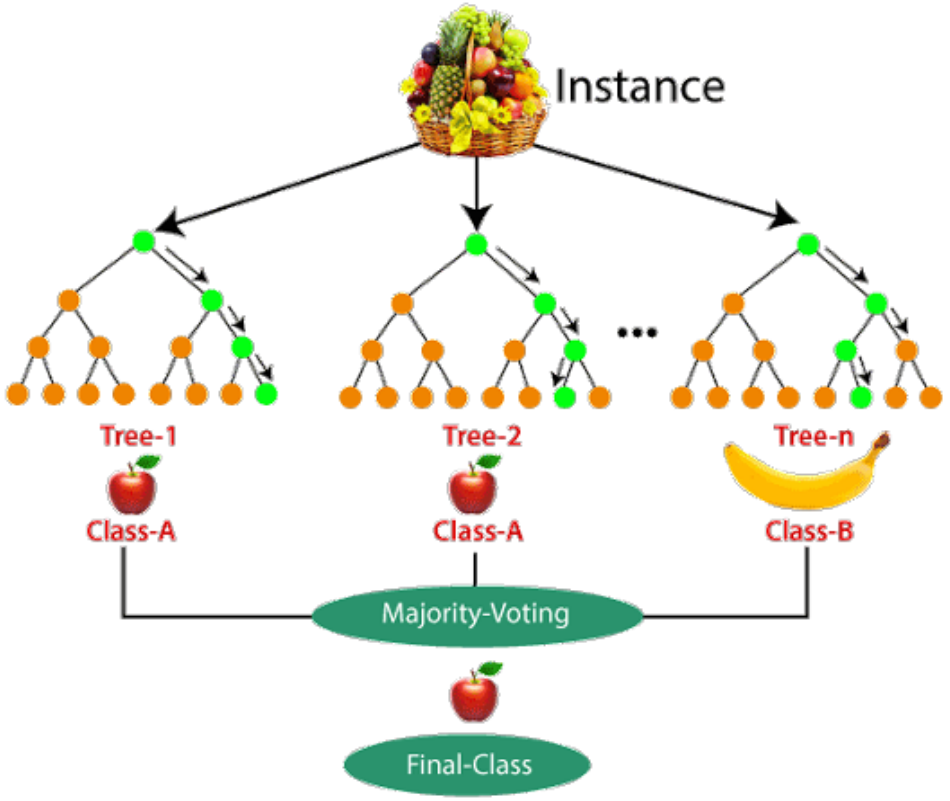


Image from: A. Singh, et. al. CanDetEnsemble: A Model for Determining and Forecasting Lung Cancer in Humans by Exploiting Various Machine Learning Algorithms

Ensemble Learning

TEXAS A&M UNIVERSITY Engineering

- Aggregate groups of predictors
  - Train classifiers and majority vote
  - Improves overall accuracy, even if component classifiers are “weak learners”
- **Boosting** is technique of combining weak multiple weak learners into a strong learning aggregate predictor (eg. AdaBoost, Gradient Boost)

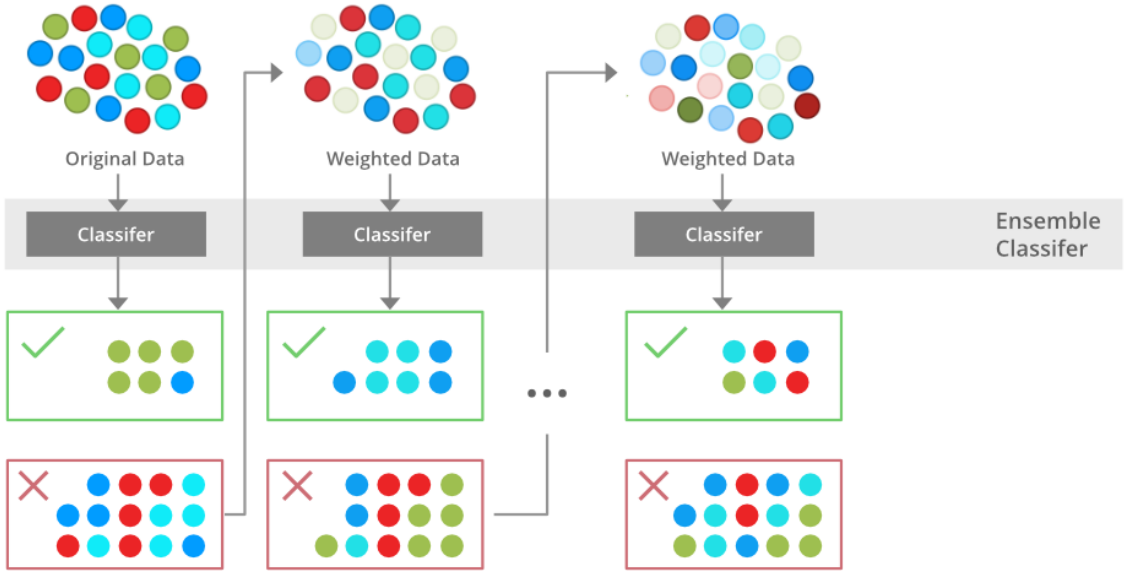


Image from: Dilaxsaswaran, Bagging vs Boosting: Understanding the Key Differences in Ensemble Learning.

Ensemble Learning

TEXAS A&M UNIVERSITY
Engineering

- Aggregate groups of predictors
  - Train classifiers and majority vote
  - Improves overall accuracy, even if component classifiers are “weak learners”
- **Bagging and Pasting**
  - Same algorithm with different subsets of training data
  - Bagging (aka Bootstrapping): w/ replacement in subset selection
  - Pasting no replacement in subset selection

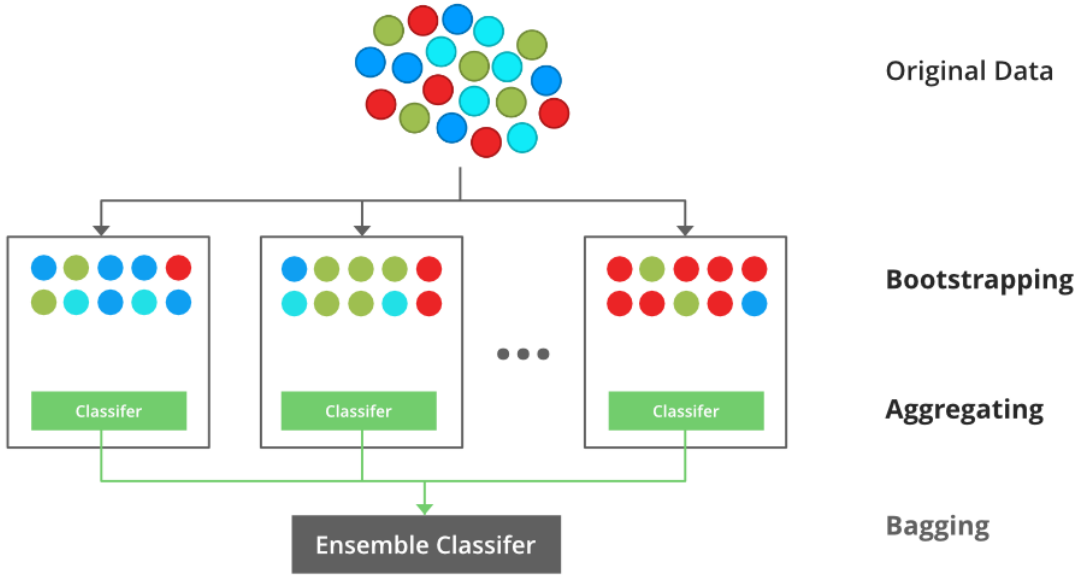


Image from: Dilaxsaswaran, Bagging vs Boosting: Understanding the Key Differences in Ensemble Learning.

Decision Tree and Random Forest Comparison

TEXAS A&M UNIVERSITY Engineering

- Both available in Sklearn!

Max Depth HP	Max Tree Depth	Train Accuracy	Validation Accuracy
1	1	0.616	0.711
2	2	0.797	0.735
3	3	0.855	0.735
4	4	0.861	0.735
5	5	0.882	0.759
6	6	0.918	0.759
none	13	0.918	0.759
Random Forest	2	0.803	0.771

Random Forest: Feature Importance

TEXAS A&M UNIVERSITY Engineering

- Aggregate groups of Decision Trees
  - Random forests measure feature importance which can influence predictor choice, drive feature selection,…feature_importances_ variable in RFC class in Scikit-Learn

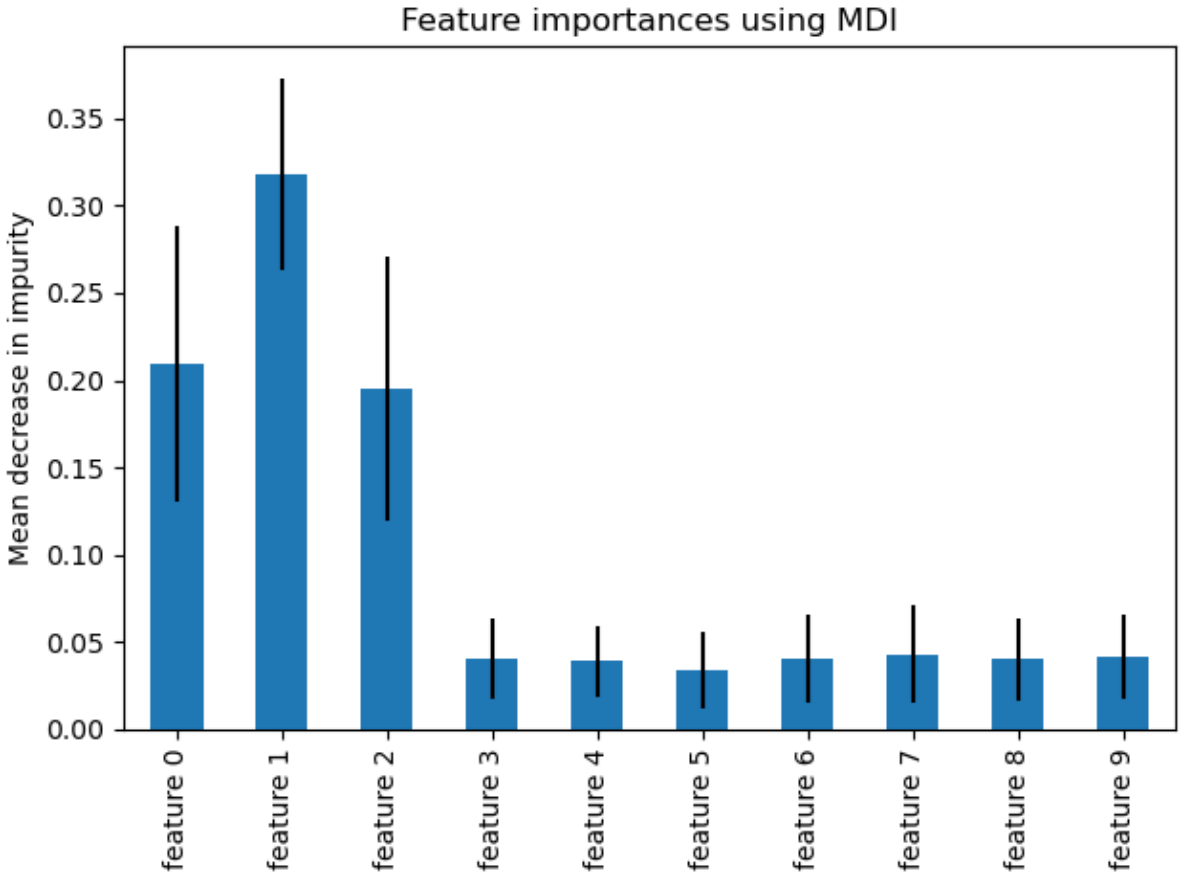


Image from: Deng, et. al., Ensemble learning for the early prediction of neonatal jaundice with genetic features.

Extreme Gradient Boosting (XGBoost)

TEXAS A&M UNIVERSITY Engineering

- Uses parallel trees for boosting
  - Unlike Random Forest that typically use bagging
- XGBoost is scalable and can be used with GPUs

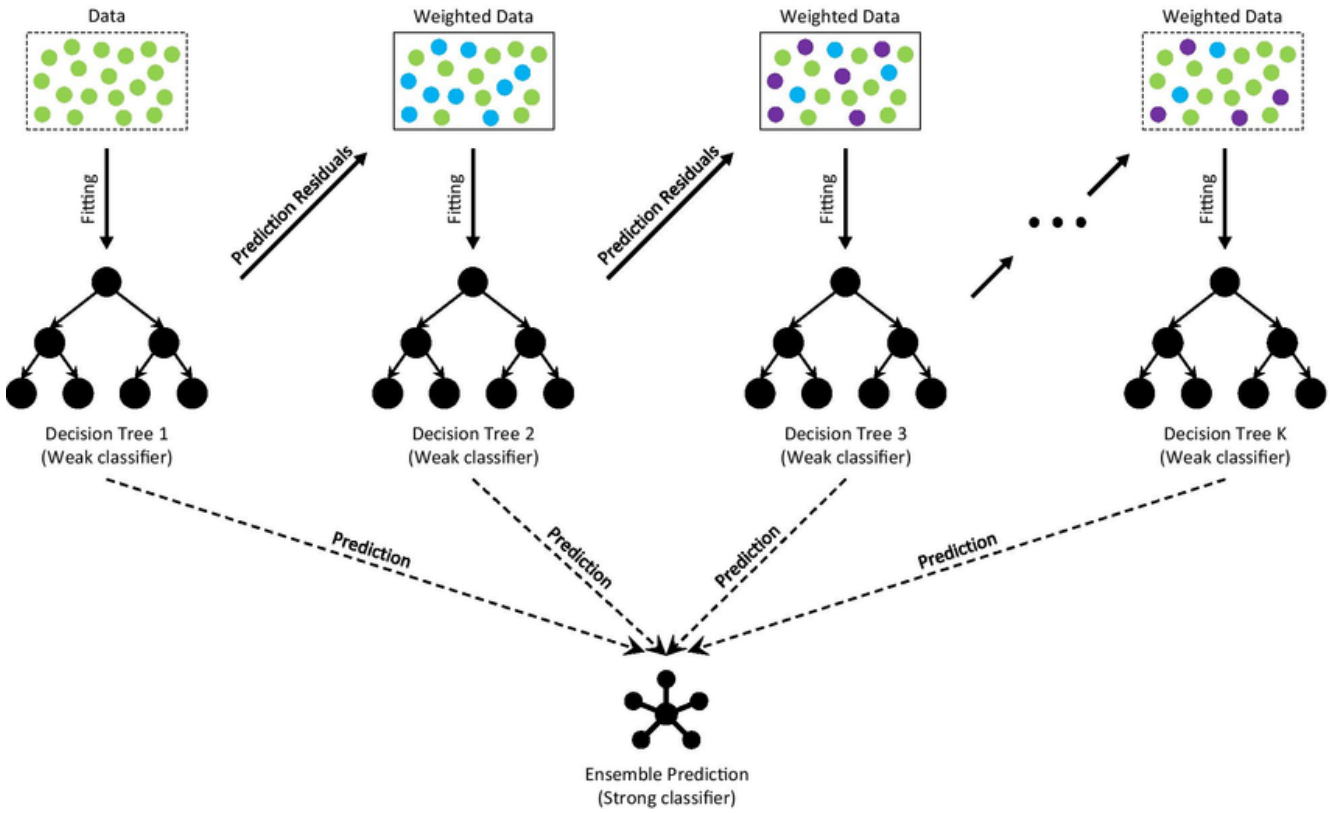
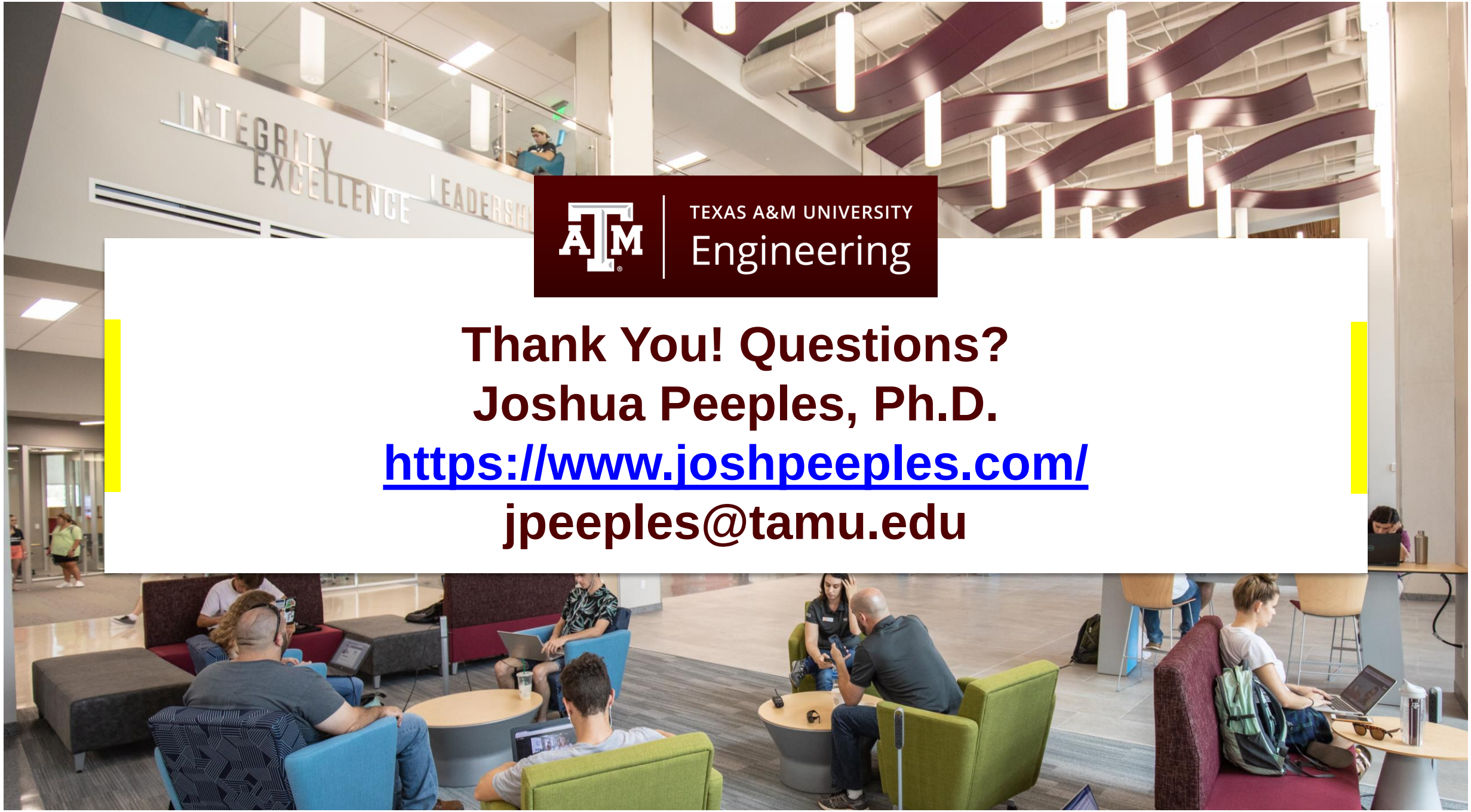


Image from: Deng, et. al., Ensemble learning for the early prediction of neonatal jaundice with genetic features.

Next class

- Guest lecture (10/09)
- Enjoy your Fall Break ☺



TEXAS A&M UNIVERSITY
Engineering

Thank You! Questions?

Joshua Peeples, Ph.D.

https://www.joshpeeples.com/

jpeeples@tamu.edu

TEXAS A&M UNIVERSITY
Engineering

Supplemental Slides

Useful Links

TEXAS A&M UNIVERSITY
Engineering

- The Basics of Decision Trees
- Decision Trees: Gini vs Entropy
- Introduction to Random Forest in Machine Learning
- Bagging vs Boosting
- XGBoost
- XGBoost vs Random Forest
- Visual Guide to XGBoost