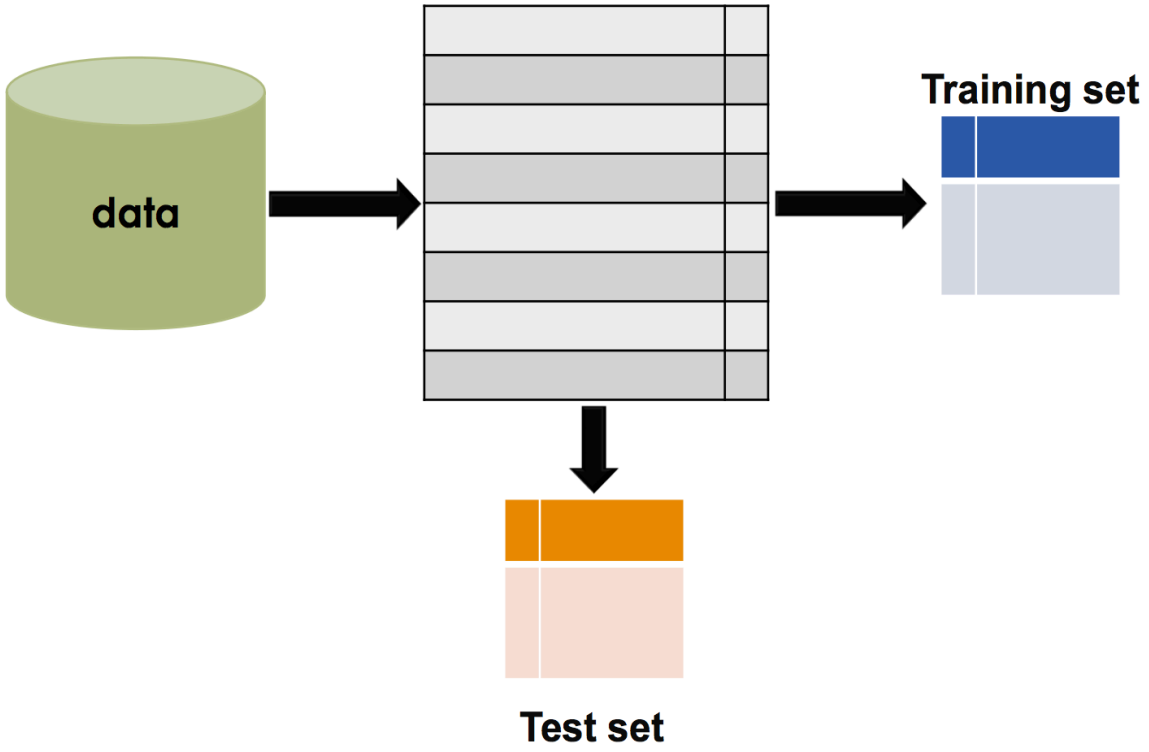

PRACTICE
OF DATA
SCIENCE

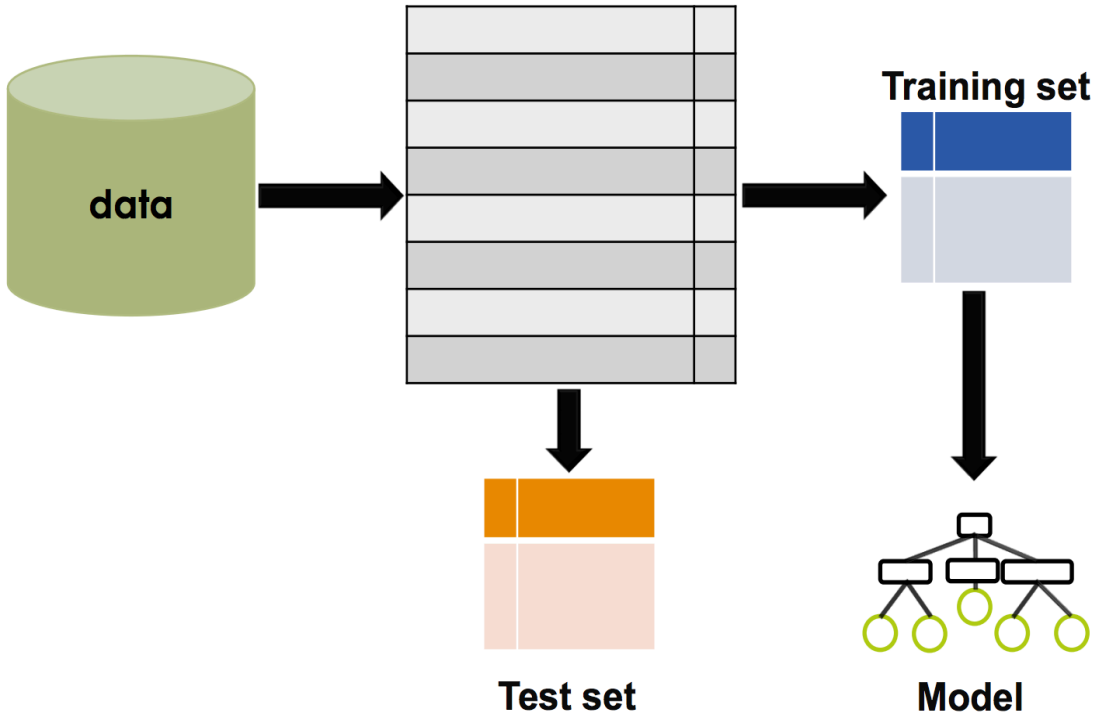
Supervised Machine Learning

Computer learns through examples how to classify events/objects/instances.



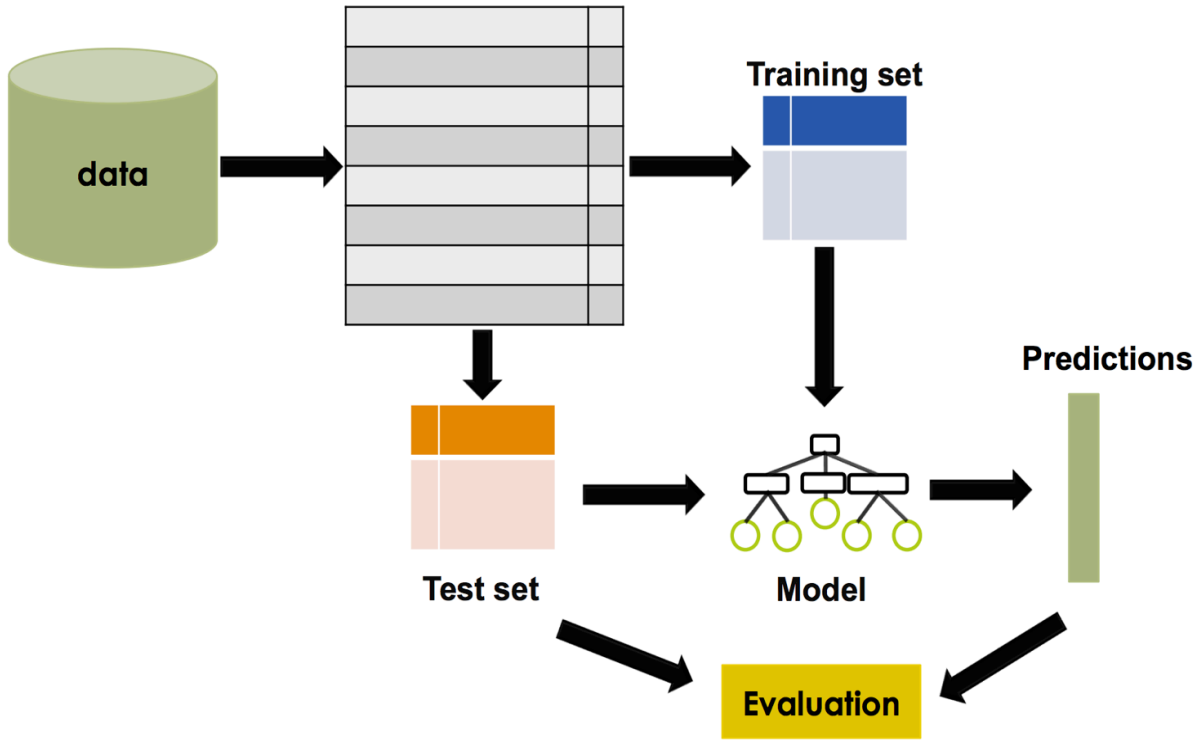
Supervised Machine Learning

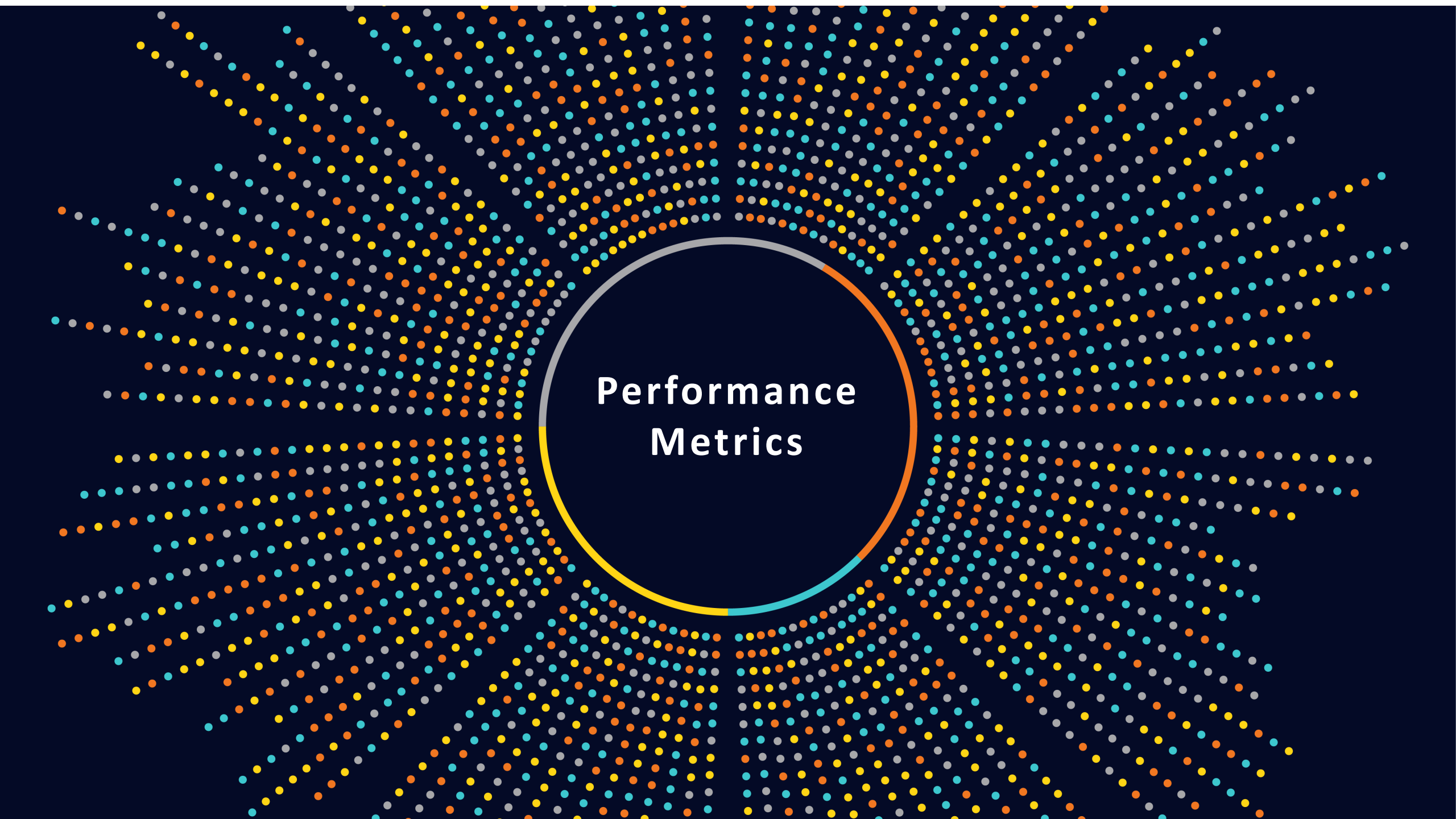
Computer learns through examples how to classify events/objects/instances.



Supervised Machine Learning

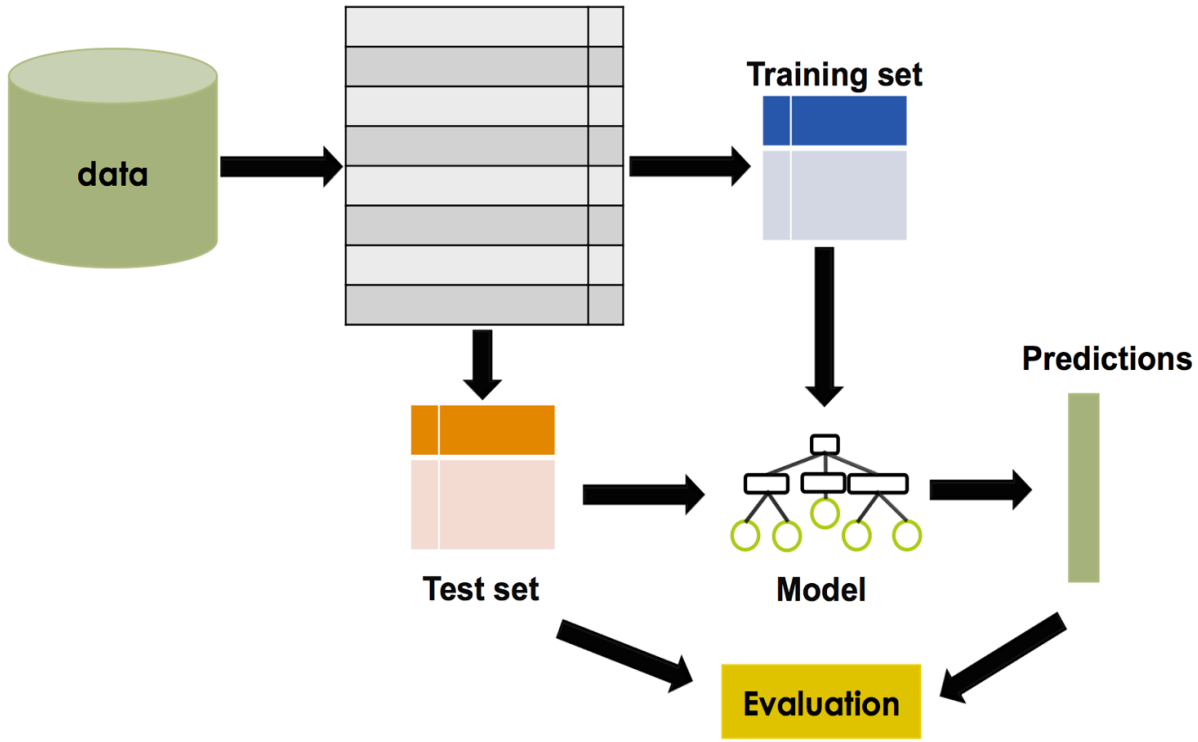
Computer learns through examples how to classify events/objects/instances.




Performance
Metrics

Supervised Machine Learning

Computer learns through examples how to classify events/objects/instances.



Evaluation of Performance

- What metrics should we use?
  - Accuracy may not be enough
- How reliable are the predicted values from your model?
- Are errors on the training data a good indication of errors on future data?
  - optimistic

True Positive (TP): • Reality: A wolf threatened. • Shepherd said: "Wolf." • Outcome: Shepherd is a hero.	**False Positive (FP):** • Reality: No wolf threatened. • Shepherd said: "Wolf." • Outcome: Villagers are angry at shepherd for waking them up.
False Negative (FN): • Reality: A wolf threatened. • Shepherd said: "No wolf." • Outcome: The wolf ate all the sheep.	**True Negative (TN):** • Reality: No wolf threatened. • Shepherd said: "No wolf." • Outcome: Everyone is fine.

Source: https://developers.google.com/machine-learning/crash-course/classification/true-false-positive-negative

- Confusion Matrix, ROC (receiver operating curve) and AUC (area under the curve) are very common approaches for measuring the performance of classification models

- Classification models output percentages that an individual input will belong to a specific class, usually a 1 or 0, with one being a positive attribute.
  - Likelihood of email spam/fraud is an example. The higher the model percentage prediction on any one email the higher chance it is fraud.

- Essentially both measure the misclassification error rate associated with your model

- A Confusion Matrix is a good tool for understanding how accurate you model is classifying and is used to build ROC

- Let's use intruder/fraud detection as an example
- Say we have 135 emails entering our system and we are trying to detect whether they are fraudulent or not
  - We use lots of criteria – source, subject, if they came from a prince…
- Generate probability measures as a result for a tree-based classifier to determine the likelihood that any one of these emails is fraudulent
- The cutoff point that is predetermined in the tree (and is a universal standard) is 50% but can be modified as an input if needed

- Below is are the results of our model in a **Confusion Matrix**. They center on the positive and negative classifications in sub-categories of true and false positive.
- Keep in mind we know because of the labels, what is fraud and not, so we can measure how good the model is classifying.
- Both true negative and true positive are good, false negative and false positive are errors.

1 = Fraud/Spam 0 = Not Fraud/Spam		Predicted Class	
		Positive Fraud Pred (1)	Negative Not Fraud Pred (0)
Actual Class	Positive Fraud Actual (1)	True Positive 10	False Negatives 22
	Negative Not Fraud Actual (0)	False Positives 7	True Negative 96

- Let's consider the extremes: what if we set the threshold to 0?
  - Means that everything is captured as Fraud and no ever gets an email again!

1 = Fraud/Spam 0 = Not Fraud/Spam		Predicted Class	
		Positive Fraud Pred (1)	Negative Not Fraud Pred (0)
Actual Class	Positive Fraud Actual (1)	True Positive 32	False Negatives 0
	Negative Not Fraud Actual (0)	False Positives 103	True Negative 0

- Let's consider the other extreme: what if we set the threshold to 100?
  - Means nothing is fraud and now everyone is getting rich off of Arabian princes

1 = Fraud/Spam 0 = Not Fraud/Spam		Predicted Class	
		Positive Fraud Pred (1)	Negative Not Fraud Pred (0)
Actual Class	Positive Fraud Actual (1)	True Positive 0	False Negatives 32
	Negative Not Fraud Actual (0)	False Positives 0	True Negative 103

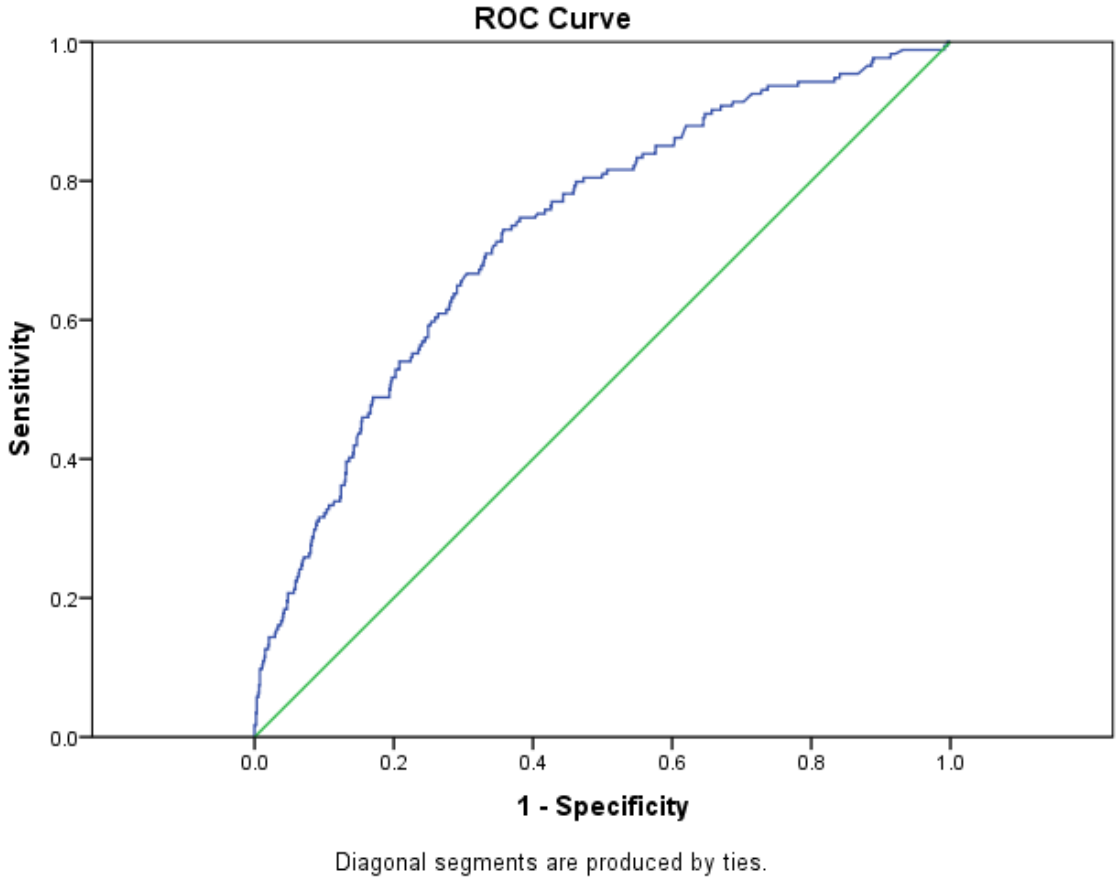
- We can further assess our model by generating classification rates:
  - True Positive Rate (TPR) (Sensitivity) = TP/(TP+FN) = 10/(10+22) = .31
    - % of fraud correctly labeled as fraud
  - False Positive Rate (FPR) (1- Specificity) = FP/(FP+TN) = 7/(7+96) = .06
    - % of emails labelled not fraud that were false positives

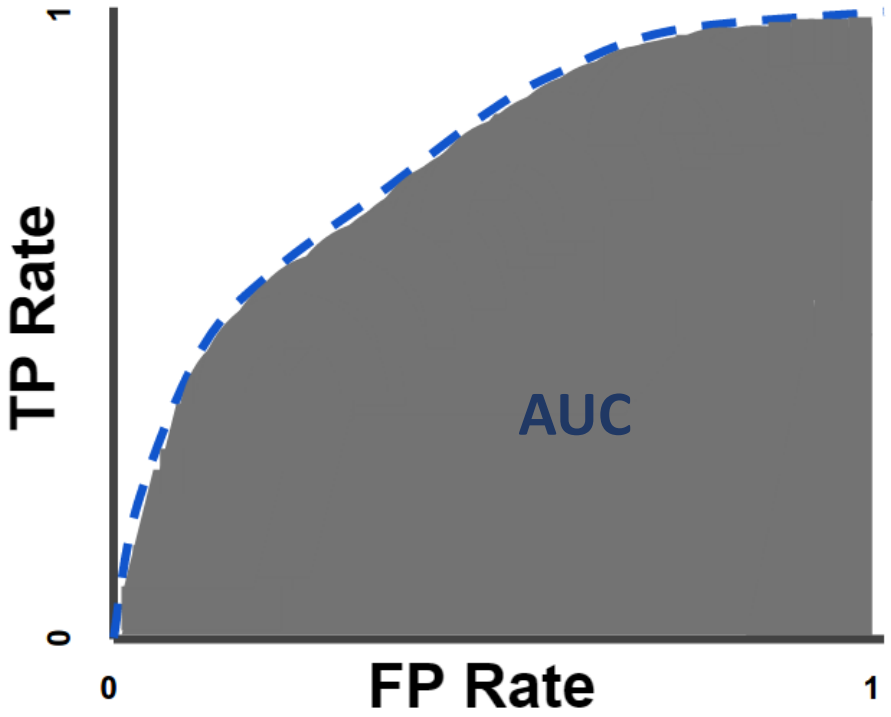
1 = Fraud/Spam 0 = Not Fraud/Spam		Predicted Class	
		Positive Fraud Pred (1)	Negative Not Fraud Pred (0)
Actual Class	Positive Fraud Actual (1)	True Positive 10	False Negatives 22
	Negative Not Fraud Actual (0)	False Positives 7	True Negative 96

- These two data points can used to begin to develop a Receiver Operating Characteristic Curve or ROC curve
  - True Positive Rate (TPR) = 10/(10+22) = .31 = y-axis
  - False Positive Rate (FPR) = 7/(7+96) = .06 = x-axis

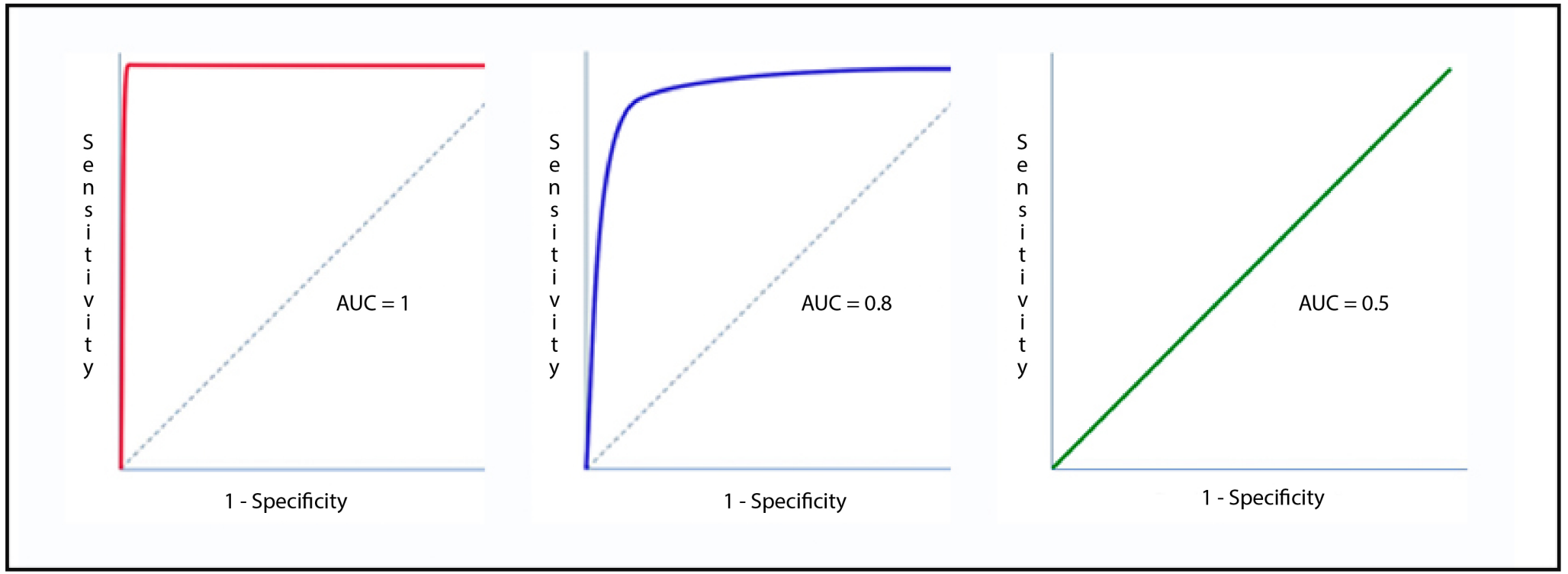
1 = Fraud/Spam 0 = Not Fraud/Spam		Predicted Class	
		Positive Fraud Pred (1)	Negative Not Fraud Pred (0)
Actual Class	Positive Fraud Actual (1)	True Positive 10	False Negatives 22
	Negative Not Fraud Actual (0)	False Positives 7	True Negative 96

- ROC curve is essentially a graphical representation of the adjusted threshold values of the confusion matrix, below are two examples





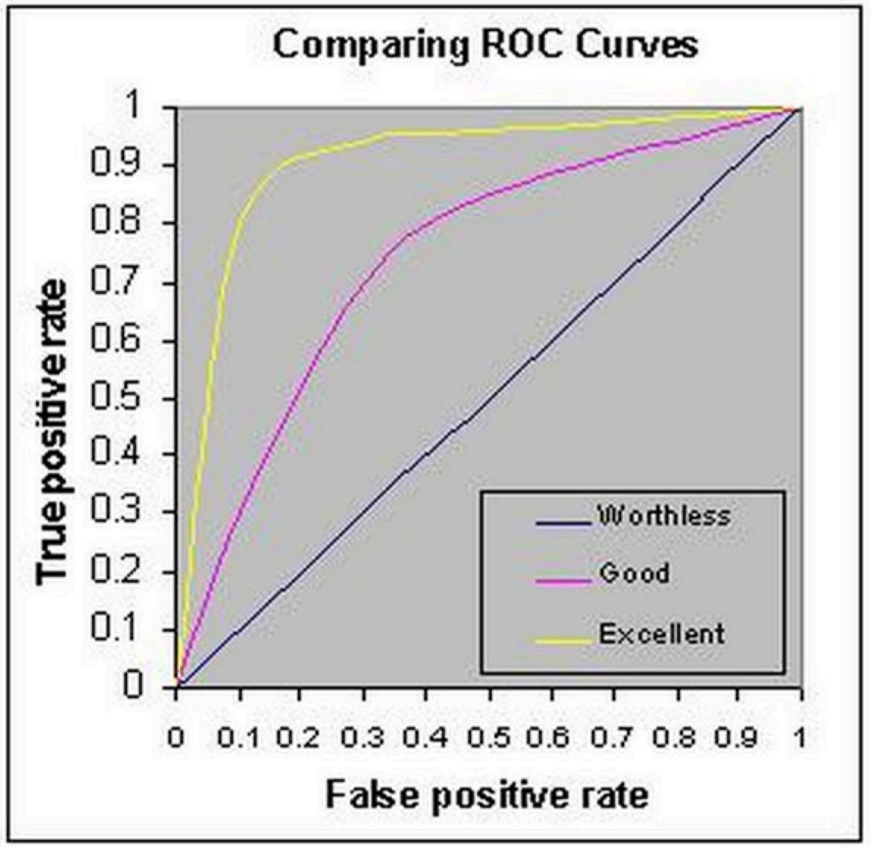
- ROC curve generates the area under the curve as a percentage of the total graph under the curve.



The Area Under the Curve (AUC) is indicative of performance.

AUC:

0.9 – 1.0 = Excellent
0.8 – 0.9 = Good
0.7 – 0.8 = Fair
0.6 – 0.7 = Poor
0.5 – 0.6 = Fail



True Positive (TP): • Reality: A wolf threatened. • Shepherd said: "Wolf." • Outcome: Shepherd is a hero.	**False Positive (FP):** • Reality: No wolf threatened. • Shepherd said: "Wolf." • Outcome: Villagers are angry at shepherd for waking them up.
False Negative (FN): • Reality: A wolf threatened. • Shepherd said: "No wolf." • Outcome: The wolf ate all the sheep.	**True Negative (TN):** • Reality: No wolf threatened. • Shepherd said: "No wolf." • Outcome: Everyone is fine.

Source: https://developers.google.com/machine-learning/crash-course/classification/true-false-positive-negative

Additional Performance Measures

True Positive (TP): • Reality: A wolf threatened. • Shepherd said: "Wolf." • Outcome: Shepherd is a hero.	**False Positive (FP):** • Reality: No wolf threatened. • Shepherd said: "Wolf." • Outcome: Villagers are angry at shepherd for waking them up.
False Negative (FN): • Reality: A wolf threatened. • Shepherd said: "No wolf." • Outcome: The wolf ate all the sheep.	**True Negative (TN):** • Reality: No wolf threatened. • Shepherd said: "No wolf." • Outcome: Everyone is fine.

Accuracy – *(TP+TN)/(TP+FP+FN+TN)*

Prevalance – The percentage of the positive class in the test data set

Detection Rate - The rate of true events also predicted to be events

Balanced Accuracy - *(sensitivity+specificity)/2*

Precision - *TP/TP+FP* – When predicting **True Positives**, what percentage is correct? (no FP, precision = 1)

Recall – *TP/TP+FN* (same as sensitivity) – What proportion of **Actual Positives** where identified correctly?

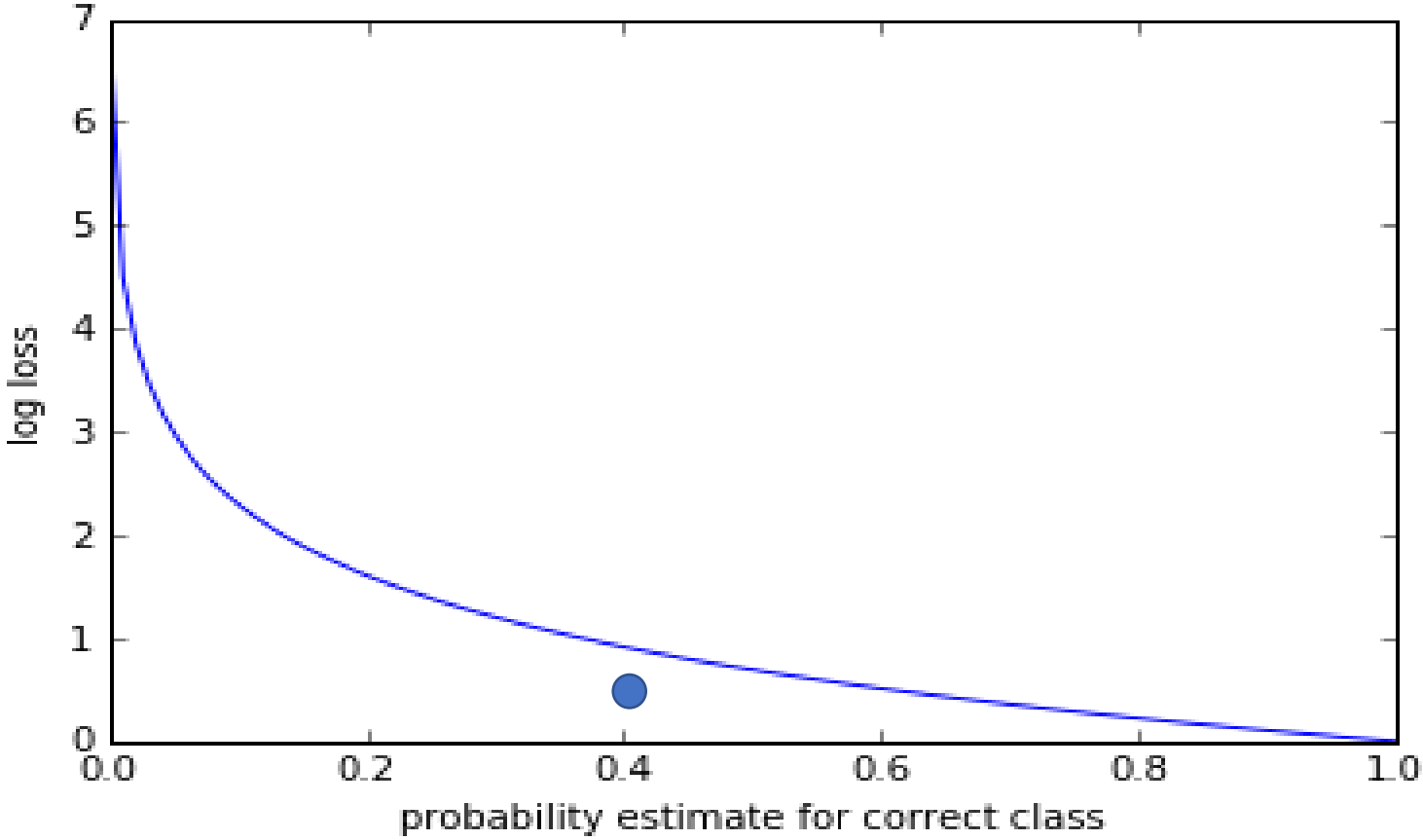
F1 Score – Harmonic mean of Precision and Recall, where accuracy is used when True Positives and True Negatives are important, F1 is used when False Negatives and False Positives are more of a concern. Also really best used on unbalanced datasets

$$\text{F1-score} = \left(\frac{\text{Recall}^{-1} + \text{Precision}^{-1}}{2}\right)^{-1} = 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}$$

Additional Performance Measures

Log Loss - log loss measures the UNCERTAINTY of the probabilities of your model by comparing them to the true labels – CLASSIFICATION. It heavily penalizes classifications that are highly confident in the wrong direction.

So, seeing a log loss of 1 can be expected in the case when our model only gives less than a ~ 40% probability estimate for selecting the actual class. Knowing the baseline rate (prevalence) here is important!



$$logLoss = \frac{-1}{N}\sum_{i=1}^{N}(y_i(logp_i) + (1 - y_i)log(1 - p_i))$$

Additional Performance Measures

Kappa - Landis and Koch (1977) provide a way to characterize values. According to their scheme a value < 0 is indicating no agreement , 0–0.20 as slight, 0.21–0.40 as fair, 0.41–0.60 as moderate, 0.61–0.80 as substantial, and 0.81–1 as almost perfect agreement.

Indicates how much better our classifier is performing over the performance of a classifier that would just guess at random according to the frequency of each class.

Its especially useful for multi-class models as many of the metrics we have reviewed are better suited to binary examples.

- These metrics can be used to assess the fairness of the machine learning models

- We will review these topics again later in the semester, as there's much more to be learning but having a basic understand will help when we walk through the fairness formulas.

Another Definition: Functional Approximation

- What is a functional approximation problem?
  - Target variable: Dependent: What we are trying to predict
  - Other Variables: Independent: Using to Predict
- Functional approximation is an approach that uses the other variables we have access to approximate the dependent and does so through the function development
- We will use regression, which assumes that we have a numeric target variable, for classification it's often a bi-variate or class level variable

Assessment Measures

- Assessing Regression Models: MSE, RMSE and MAE and Log Loss
  - MSE – The difference between the predicted values and the actual values squared
  - RSME – Same as above only the square root is taken to put the error back in terms of the dependent variable
    - Can also normalize the RSME to the range of the data in order to be able to compare RSME outputs that include different data ranges
  - MAE – The same approach only taking the absolute value instead of squaring

Equations

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

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|x_i - x|$$

$$NRMSE = \frac{RMSE}{X_{obs,\max} - X_{obs,\min}}$$

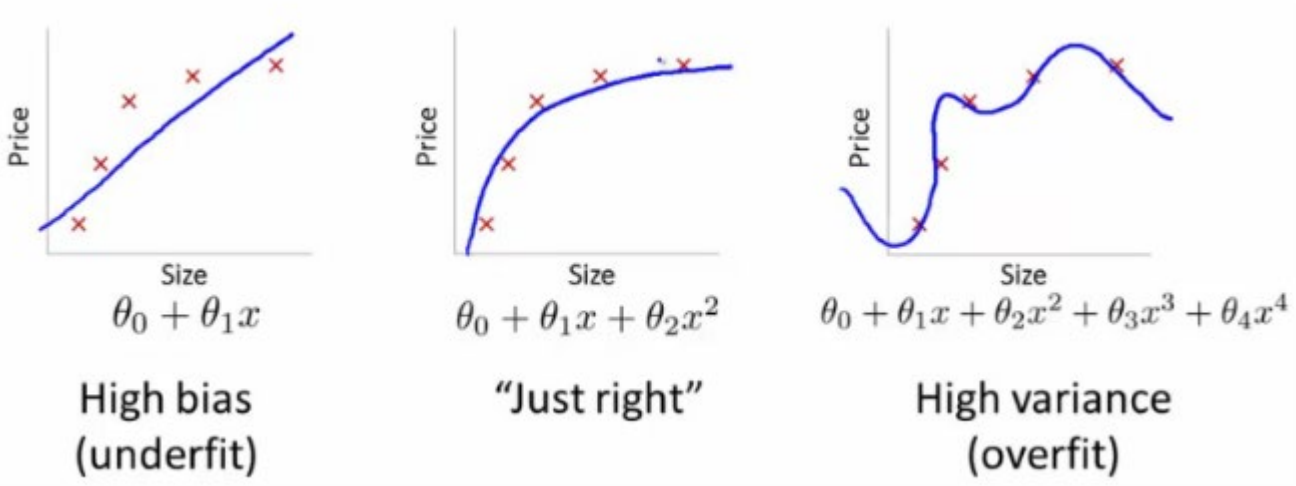
$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(X_{obs,i} - X_{model,i})^2}{n}}$$

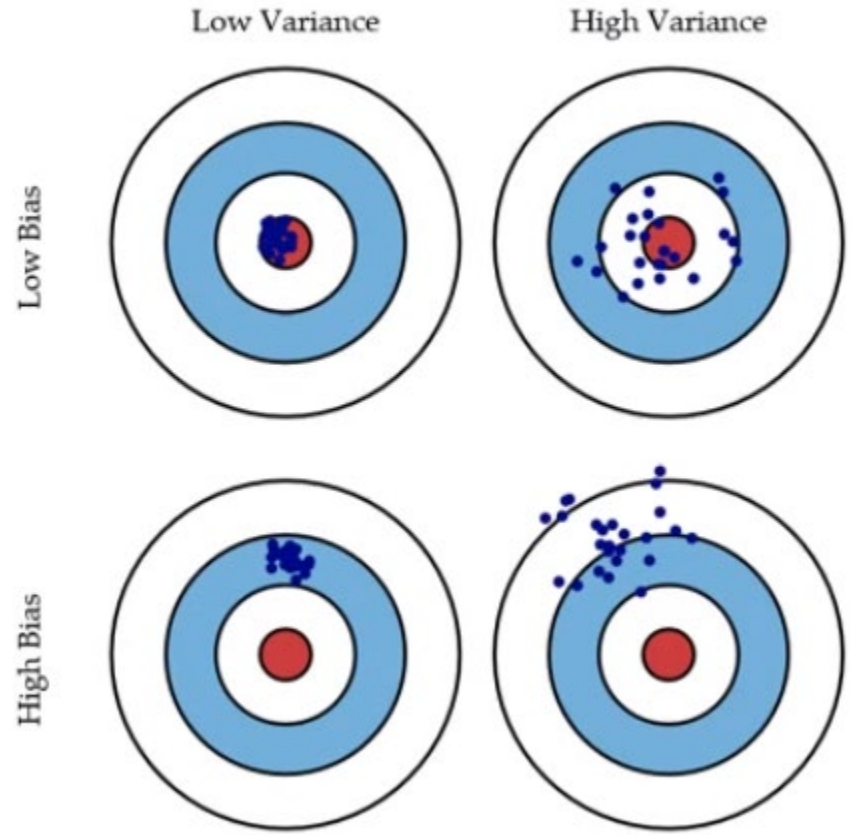
Regression

- Said another way basic linear regression has a Prediction Accuracy Problem:
  - Has a low bias (overfitting) but a high variance
  - This can be improved by injecting some level of bias into the equation by reducing the impact of certain coefficients
  - This can improve overall accuracy by reducing variance
- Draw Dart Board –
- Another issue is interpretation – with a large number of predictor variables and large data sets it often hard to identify variable importance and explain model outcomes

Regression and Sparsity

- Often we have more features than observations in the world of big data
  - What type of problem is this?
- So we strive to have Sparse models
  - What do we mean by Sparse?


Price
Size
$\theta_0 + \theta_1 x$
High bias
(underfit)
Price
Size
$\theta_0 + \theta_1 x + \theta_2 x^2$
"Just right"
Price
Size
$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$
High variance
(overfit)


Low Variance
High Variance
Low Bias
High Bias

The Confusion Matrix and Accuracy

		Predicted Class	
		Yes	No
Target Class	Yes	True positives	False negatives
	No	False positives	True negatives

$$Accuracy = \frac{\#\ correctly\ classified\ tuples}{total\ \#\ tuples}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy

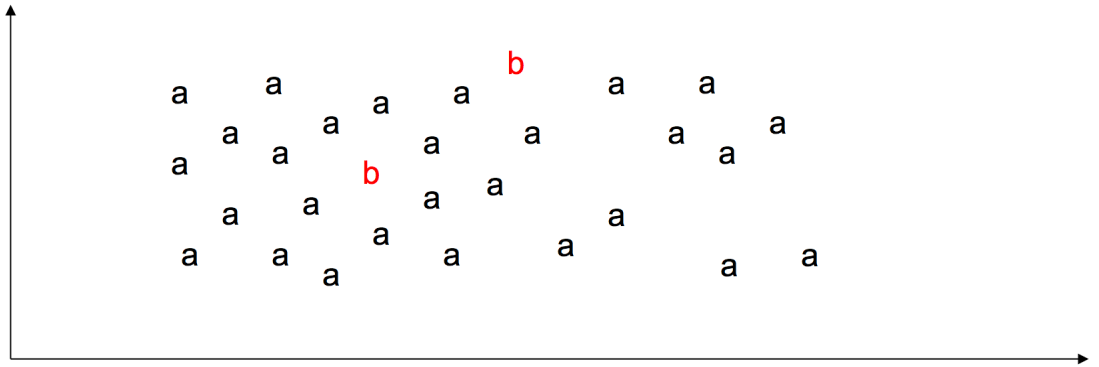
Most widely used metric

- Accuracy is better measured using test data that was not used to build the classifier.
- Referred to as the overall recognition rate of the classifier
- Error rate or misclassification rate:

  1-Accuracy
- When training data are used to compute the accuracy, the error rate is called resubstitution error.

Consider this two-class case.

Accuracy may not be enough.



Consider the accuracy if a model predicts that the outcome is always class a.

28/30 = 93% accuracy

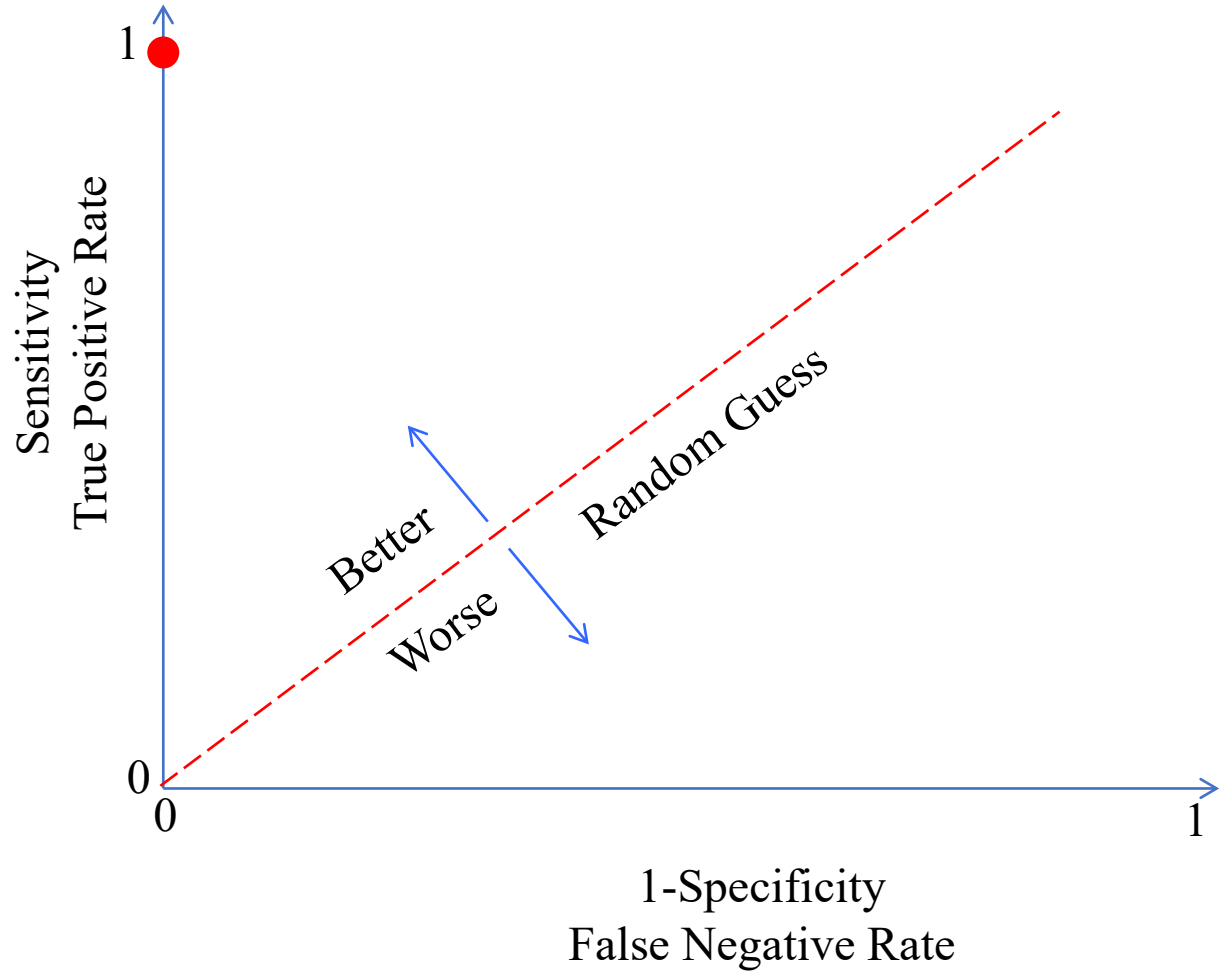
This is misleading.

Sensitivity and Specificity

$$Sensitivity(sn) = \frac{TP}{\#\,Positives}$$

$$Specificity(sp) = \frac{TN}{\#\,Negatives}$$

Receiver Operating Characteristic (ROC) Curve



The ROC Curve

T4 value	Hypothyroid	Euthyroid
5 or less	18	1
5.1 - 7	7	17
7.1 - 9	4	36
9 or more	3	39
Totals:	32	93

Cutpoint	True Positives	False Positives
5	0.56	0.01
7	0.78	0.19
9	0.91	0.58

Cutpoint	Sensitivity	Specificity
5	0.56	0.99
7	0.78	0.81
9	0.91	0.42