Week 8 - 05/21/2021

CSM148 Discussion Section 1B

The week of reckoning is upon us!

- Because I couldn't give you any memes yesterday...

- Midterm This Thursday!
- **Project 3 due in 10 days (June 1)**
- HW3 due in 2 weeks (June 3)

Midterm Scope Study Guide

- Learn a Model
  - You may be asked to leverage a learning rule to create or update parameters on a model. Examples we've covered in class that you will be expected to be able by how to solve by hand are:
    - Perceptrons
    - KMeans
- Generate Predictions
  - You may be asked to leverage an existing model to generate a series of predictions. Examples we've covered in class that you will be expected to be able by how to solve by hand are:
    - Linear Regression
    - KNN
    - Naive Bayes
    - Neural Net
    - SVM

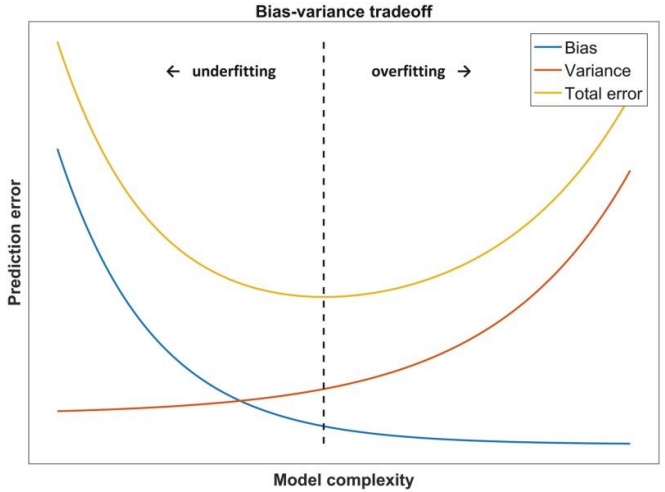
Midterm Scope Study Guide

- Metrics Generation/Interpretation
  - You may be asked to generate or interpret metrics associated with classification or regression. Please be certain to be familiar with the following:
    - Loss functions (MSE, RMSE, MAE)
    - Statistical metrics (P-values, T-Values, R2)
    - Classification Metrics (Accuracy, Precision, Recall, F1 Score)
    - Confusion Matrixes
    - ROC/AUC
- Comparative Assessment
  - Much of the work of a data scientist involves evaluating basic tradeoffs between options available to them. You may be asked about a specific scenario and asked to evaluate how different learning models may handle it, how different parameters might impact the performance, or how different tools at your

Other elements in scope

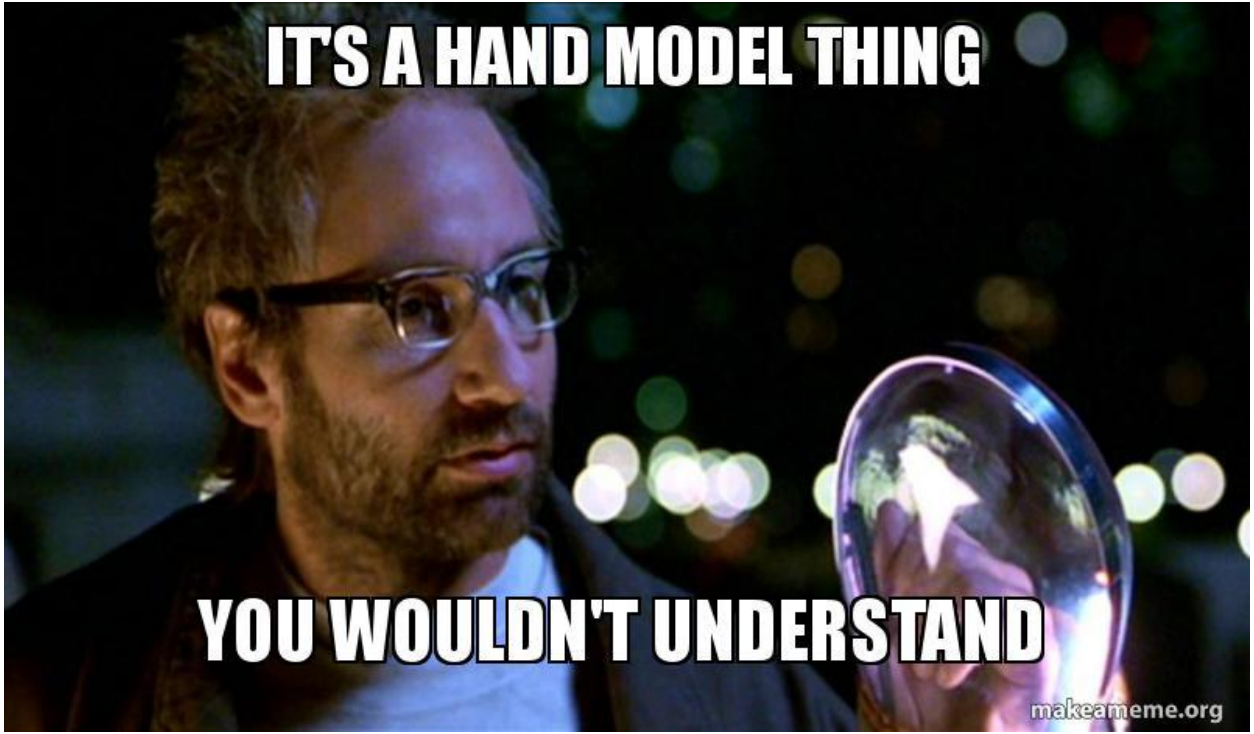
- Modeling tools
  - Data processing tools
    - Imputation, categorical encoding, scaling
    - PCA
    - Data splitting/crossvalidation
  - Parameter Optimizations
    - Loss functions
    - Normalization techniques
    - Gradient Descent
- Models
  - Regression
    - Linear
    - KNN
  - Classification
    - Logistical
    - SVM
    - Bayesian
    - KNN
    - Decision Tree
    - Random Forest
    - Ensemble methods
    - Perceptron
    - Neural Net
    - KMeans

- Questions to be prepared to answer:
  - How does this particular method ‘learn’ a model of the data
  - How is a prediction applied (i.e., how are decision boundaries determined)
  - What are the relative strengths and weaknesses of each approach
  - What tools does each model have to try and optimize the Variance/Bias tradeoff



Model Solving

Model Solving



- KMeans
- Perceptron

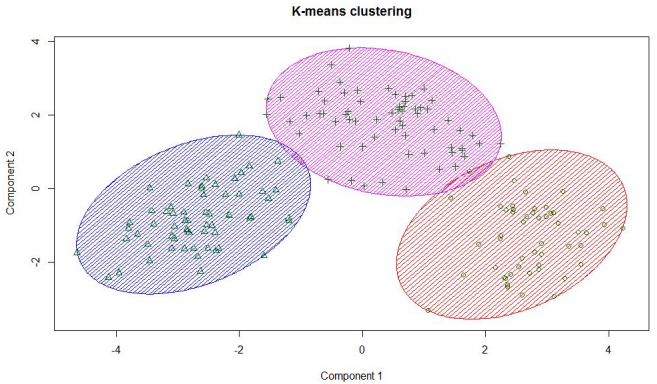
KMeans

KMeans

- KMeans is one of the most basic and intuitive Clustering algorithms
- Kmeans iteratively tries to partition the dataset into K, distinct, non-overlapping, clusters

The algorithm works as follows:

1. First we initialize k points, called means, randomly.
2. We categorize each item to its closest mean and after doing so for all points, we update the mean's coordinates, based on the items categorized in that mean so far.
3. We repeat the process for a given number of iterations, until the points in the cluster do not change, at which point our model has converged.



KMeans Example

1. Define K = 2
2. Assign centroids at random: In this case opposite ends (1,1) and (5,7)
3. Iterate through all data, assign to the nearest cluster
4. Recalculate the centroids as the mean value of all assigned points
5. Now recalculate the distance between each point and the new centroid value to see if any of the points are closer to the other cluster.
    a. In our case, after recalculating our distances. ID 3 needs to be reassigned, and the centroids are recalculated.
6. If any points are reassigned, recalculate the new centroid and reassess the values
7. Keep running until the model has converged and no further realignment occurs

ID	X	Y
1	1	1
2	1.5	2
3	3	4
4	5	7
5	3.5	5
6	4.5	5.0
7	3.5	4.5

Epoch	Centroids	Cluster 1	Cluster 2
0	(1,1), (5,7)	1, 2, 3	4, 5, 6, 7
1	(1.83, 2.33), (4.1, 5.4)	1, 2	3, 4, 5, 6, 7
2	(1.25, 1.5), (3.9, 5.1)	1, 2	3, 4, 5, 6, 7

Perceptron

Perceptron Model

- Each neuron takes inputs
  - Weighs them separately,
  - Sums them
  - Passes this sum through a nonlinear ‘activation’ function to produce output.
    - Activation functions are used to map the input between the required values like (0, 1) or (-1, 1).

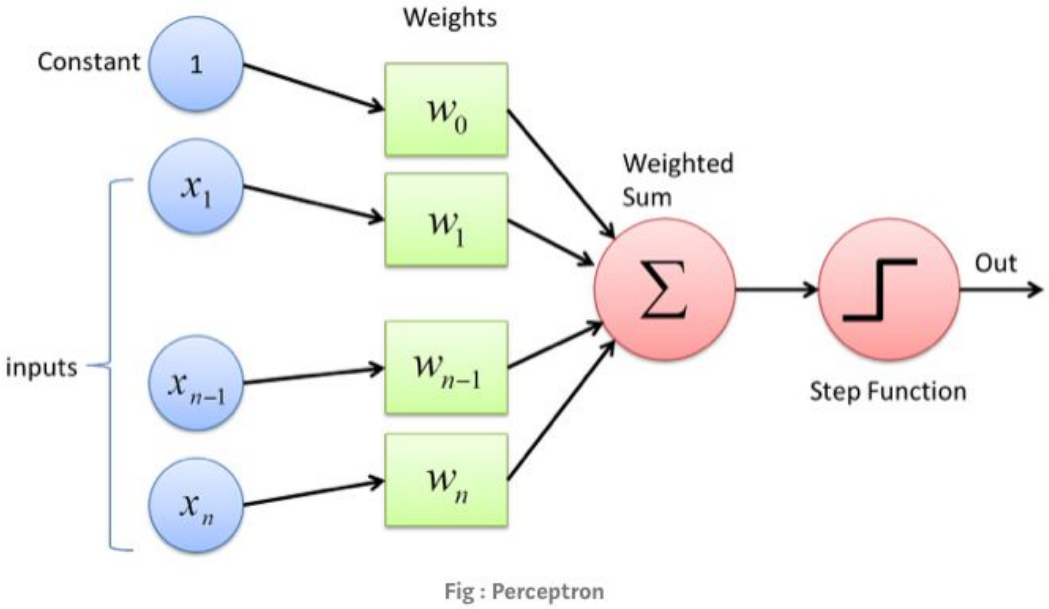


Fig : Perceptron

$$\sum_{i=1}^{m} w_i x_i$$

$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

How Perceptrons 'Learn'

$$\Delta w_i = c(t - z) * x_i$$

- Where **Wi** is the weight from input i to perceptron node, **C** is the learning rate, **t** is the target for the current instance, **Z** is the current output, and **Xi** is 'i'th input
- We apply a least perturbation principle
  - Only change weights if there is an error
  - Use a small c rather than changing weights sufficient to make current pattern correct
  - Scale by the value of Xi - Important!!!!

Steps to run a Perceptron:

1. Create a perceptron node with n inputs
2. Iteratively apply a pattern from the training set and apply the perceptron rule
3. Each iteration through the training set is an epoch
4. Continue training until total training set error ceases to improve

Perceptron Convergence Theorem: Guaranteed to find a solution in finite time if a solution exists

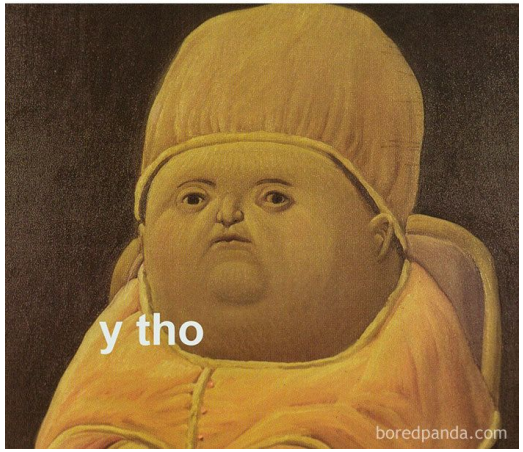
Weight 'Training'

- We have a perceptron with 3 inputs and outputs a 1 if the sum is greater than 0, else it returns a 0.
- The weights for both inputs along with the constant are initially set equal to 1.
- The Learning Rate for the model is equal to 1.
  - W1X1 + W2X2 + W3X3 + W0

$$\Delta w_i = c(t - z) * x_i$$

X1	X2	X3	Actual
-8	3	4	1
1	2	1	0
3	2	3	0

When you've been dieting for 3 hours but you're still not skinny



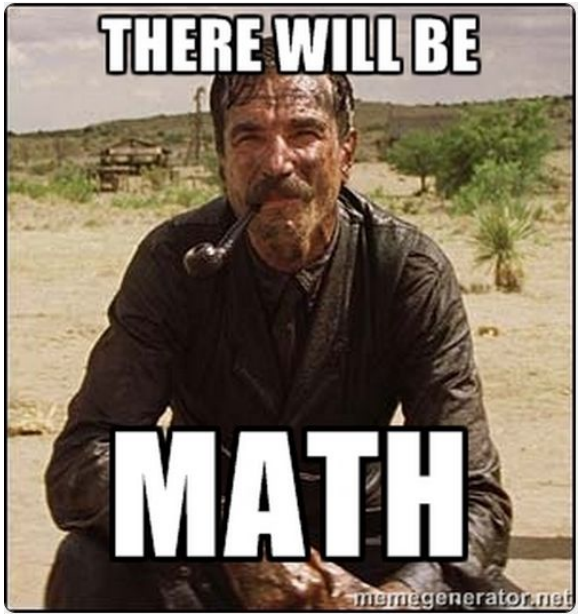
W1	X1	W2	X2	W3	X3	W0	Sum	Predict	Actual
1	-8	1	3	1	4	1	0	0	1
1-8 = -7	1	1+3 = 4	2	1+4 =5	1	1+1 =2	8	1	0
-7-1 =-8	3	4-2 =2	2	5-1 =4	3	2-1 =1	-7	0	0

Metrics

Model Evaluation

- Calculate:
  - Accuracy
  - Precision
  - Recall
  - F1 Score

n = 165	Predicted: No	Predicted: Yes
Actual: No	50	10
Actual: Yes	5	100



Model Evaluation

- Classification Rate/Accuracy:
  - Accuracy = (TP + TN) / (TP + TN + FP + FN)
  - = (100+50) /(100+5+10+50)= 0.90
- Recall:
  - Recall=TP / (TP + FN)
  - =100/(100+5)=0.95
- Precision:
  - Precision = TP / (TP + FP)
  - =100/ (100+10)=0.91
- F-measure:
  - Fmeasure = (2*Recall*Precision)/(Recall+Precision)
  - =(2*0.95*0.91)/(0.91+0.95)=0.92

n = 165	Predicted: No	Predicted: Yes
Actual: No	50	10
Actual: Yes	5	100

Final Thoughts

- Wednesday Night's online office hours will be reserved for exam review
- You'll do great!
- Exercise self-care
  - Get good sleep
  - Eat before the exam