**CS 4320 Lecture 2 (9/11/2020):**

* Definition of supervised learning
* Prediction
  + There is some process in the world that you want predict
  + Decide on factors that you will be using
  + Ex- movies
    - Action, euro, romance, sci-fi, 3d, result
  + Can start to train based on what movies you have like in the past based on inputs and outputs
  + Apply the learning algorithm to the training data
* Variants of supervised learning
  + Binary classification: Y=(-1,+1)
  + …
* Supervised learning
  + Task:
    - Learn to imitate a target function f:X->Y
  + Training ex S((x1,y1),…,(xn,yn))
    - Learning alg is given correct y val based on inputs x
  + Goal:
    - Find a function h: X->Y that approximates f:X->Y as well as possible
* K-Nearest Neighbor
  + Ex-movies
  + A screenshot of a cell phone

    Description automatically generated
  + Prediction instance looks like a training instance so use that prediction
  + Also look for the most similar training example
  + Check how many features match within the training data with the prediction instance and use that for the prediction
    - Alt: take some weighting ofr the labels on the training as your final answer
  + A screenshot of a cell phone

    Description automatically generated
  + Look at the training set find the k most similar entries and use the most common
  + How to handle conflict between nearest neighbors in terms of the output
    - In case of a tie go with the more similar?
    - Take a weighted majority vote
  + Weighted K-NN for Regression
    - Take the weighted averaged of the training based on the similarity
  + Types of attributes
    - Categorical
      * Eyecolor(blue, gree, brown)
    - Boolean
      * (true, false)
    - Numerical
      * Integer age: [0,105]
      * Real: height
    - Structured
      * NLP: parse tree
      * Protein: sequence of amino acids
  + Similarity measures
    - A screenshot of a cell phone

      Description automatically generated
  + Ex- Expensive housing >$200 per sq ft
    - Two features: longitude and latitude
    - Simple algorithm
    - Need similarity measure and features that are informative of target function
    - Requires large memory for large training sets
    - Requires much computation when computing a prediction
    - Curse of dimensionality
  + Ex- Collaborative filtering
    - Use user rating matrix for many users
    - Predict the star rating of user 1 based on other user’s ratings of the movie who are similar in their other ratings to user 1