Supervised

1. Decision Stump
   Total Run Time: $O(ndk)$ (k thresholds for each of d features)
   `PSEUDO CODE HERE with O(n) on steps`
2. Decision Tree
   `PSEUDO CODE HERE with O(n) on steps`
3. Naive Bayes
   Bayes Rule: $p(y_i|x_i) = \frac{p(x_i|y_i) \cdot p(y_i)}{p(x_i)}$
   `PSEUDO CODE HERE with O(n) on steps`
4. KNN
   - Non-Parametric
   - fit cost: store $O(nd)$
   - Prediction Cost: $O(nd)$
   - depends on norm
   - Usage:

   `PSEUDO CODE HERE with O(n) on steps`
5. Linear Regression
6. Non-linear Regression (Supervised)

Unsupervised

1. K-Means (Unsupervised)
   - Parametric (k, W weights)
   - fit cost: store $O(ndk)$
   - update: $O(nd)$
   - Prediction Cost: $O(nd)$
   - depends on initialization
   - Usage: convex clustering, vector quantization, Outlier Detection

   `PSEUDO CODE HERE with O(n) on steps`
2. DBSCAN (Unsupervised)
   - Non-Parametric (ϵ, MinNbr)
   - fit cost: store $O()$
   - update: $O()$
   - Prediction Cost: $O()$
   - problem: different densities
3. Outlier Detection Methods
   - Model-based methods
   - Graphical approaches (scatter plot)
   - Cluster-based methods (k-means DBSCAN)
   - Distance-based methods (KNN)
   - Supervised-learning methods
4. Recommender?

Something

1. Golden Rule:
   Test data must not influence int the training phase in any way.
2. Fundamental Trade Off:
   $E_{test} = (E_{test} - E_{train}) + E_{train}$
   E_{appx} gets smaller as n larger, grows as model gets complicated
3. Ensemble Methods
4. Cross-Validation
   k-fold CV runtime: $O()$
   `PSEUDO CODE HERE with O(n) on steps`
5. Definitions
6. Linear Algebra Notes
   Basics
   - $w^T x_i = \sum_{j=1}^{d} w_j x_{ij}$, x_i, w is d x 1
   - $a^T A b = b^T B^T a$ both sides are vectors
   - $\frac{1}{2}\|Xw - y\|_2^2 = \frac{1}{2}\sum_{i=1}^{n}(w^T x_i - y_i) = \frac{1}{2} w^T X^T X w - w^T X^T y + \frac{1}{2} y^T y$
   - ∇const $= 0, \nabla w^t b = w, \nabla \frac{1}{2} w^T A w = Aw$ if A symmetric
   - $\nabla \frac{1}{2}\|Xw - y\|_2^2 = X^T X w - X^T y$
   - Normal equation $X^T X w = X^T y$
   - $(Xw - y)^T V (Xw - y) = \sum_{i=1}^{n} v_i (w^T x_i - y_i)^2$

   Run Time
   - $X^T y : O(nd)$
   - $X^T X : O(nd^2)$
   - solve d x d system of equations : $O(d^3)$
   - solve normal equation : $O(d^3 + nd^2)$

   Gradient Descent
   - $w^{t+1} = w^t - \alpha^t \nabla f(w^t) = w^t - X^T(Xw^t - y)$ (least square)
   - cost $O(nd)$ no need to form $X^T X$
   - total cost $O(ndt)$
   - faster for large d, works generally
7. Multivariable Calc Notes
8. Probability Notes