Similarities/Differences Between algorithms:

• Decision Trees (Decision Stumps, etc.)
• Linear Regression (gradient descent)
• Polynomial Regression (and other feature expansions)
• The Perceptron
• Logistic Regression
• Neural Networks
• 1 Nearest Neighbor (1-NN)
• k Nearest Neighbors (k-NN)
• Ensemble Methods, AdaBoost
• Support Vector Machines (SVMs)
• Naive Bayes

• How will these algorithms perform on various datasets?
• What will decision boundaries look like?
• Global vs local optima during learning?
Supervised Learning:

- Difference with Unsupervised Learning, & Reinforcement Learning
- Machine Learning algorithms $\langle P,T,E \rangle$
- Classification Vs. Regression
- Error, Accuracy, Confusion Matrix, Precision, Recall

Theory:

- Training vs. Testing, Generalization
- Overfitting, Underfitting
- Bias vs. Variance
- Cross Validation
- Shattering, VC Dimension
Perceptron, Linear Regression, Logistic Regression:
  - Regularization (relation to bias and variance)
  - Learning, gradient descent
  - Loss functions
  - Features

Neural Networks
  - Structure (layers)
  - Activation functions (linear, logistic)
  - Backpropogation

Decision Trees (working knowledge):
  - Splitting
  - Entropy / Conditional Entropy
  - Information Gain
  - Pruning (deep vs shallow / stumps)
  - Overfitting
Ensembles

- AdaBoost (bias / variance, when do you stop training)
- Properties of individual learners (what can they be, high/low accuracy)

Support Vector Machines

- What are support vectors?
- margin (how does changing support vectors change decision boundary/margin)

Naive Bayes

- Basic probability, Bayes Rule
- Prior, likelihood, posterior
- Independence assumption and why does it help?
**Reference Page**

**Decision Trees** Let $D$ be the data, $C$ be the class attribute, and $A$ be an attribute (which could be $C$). The accessor $A.values$ denotes the values of attribute $A$.

$$\text{Info}(D) = \sum_{i \in C.values} -\frac{|D_i|}{|D|} \cdot \log \left(\frac{|D_i|}{|D|}\right)$$

$$\text{Info}(A, D) = \sum_{j \in A.values} \frac{|D_j|}{|D|} \cdot \text{Info}(D)$$

$$\text{Gain}(A, D) = \text{Info}(D) - \text{Info}(A, D)$$

$$\text{GainRatio}(A, D) = \frac{\text{Gain}(A, D)}{\text{SplitInfo}(A, D)}$$

$$\text{SplitInfo}(A, D) = \text{Info}(D)$$

considering $A$ as the class attribute $C$.

$$= \sum_{i \in A.values} -\frac{|D_i|}{|D|} \cdot \log \left(\frac{|D_i|}{|D|}\right)$$

**Linear Regression**

$$\min_{\theta} \frac{1}{2n} \sum_{i=1}^{n} (h_\theta(x_i) - y_i)^2 + \lambda \sum_{j=1}^{d} \theta_j^2$$

**Perceptron Update Rule**

$$\theta \leftarrow \theta + y_i x_i \quad \text{if } x_i \text{ is misclassified}$$

**Logistic Regression**

$$P(y = 1 | x) = \frac{1}{1 + \exp(-\theta^T x)} \quad \text{and} \quad P(y = 0 | x) = 1 - P(y = 1 | x)$$

$$\min_{\theta} - \sum_{i=1}^{n} [y_i \log P(y_i = 1 | x_i) + (1 - y_i) \log P(y_i = 0 | x_i)]$$

**Neural Networks**

$$\min_{\Theta} -\frac{1}{n} \left[ \sum_{i=1}^{n} \sum_{k=1}^{K} y_{ik} \log (h_\Theta(x_i))_k + (1 - y_{ik}) \log (1 - (h_\Theta(x_i))_k) \right] + \frac{\lambda}{2n} \sum_{l=1}^{L-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_l} (\theta^{(l)}_j)^2$$

**AdaBoost**

$$\beta_t = \frac{1}{2} \ln \frac{1 - \epsilon_t}{\epsilon_t} \quad \text{where} \quad \epsilon_t = \sum_{i=1}^{n} w_t(x_i) [y_i \neq h_t(x_i)]$$

$$w_{t+1}(x_i) = \frac{w_t(x_i) \exp(-\beta_t y_i h_t(x_i))}{Z_t}$$

where $Z_t = \sum_{i=1}^{n} w_t(x_i) \exp(-\beta_t y_i h_t(x_i))$

$$H(x) = \text{sign} \left( \sum_{t=1}^{T} \beta_t h_t(x) \right)$$

**Support Vector Machines**

Primal: $$\min_{w, b} \frac{1}{2} ||w||^2 \quad \text{s.t.} \quad \forall i \quad y_i (w \cdot x - b) \geq 1$$

Dual: $$\min_{\alpha} \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{n} \alpha_i \quad \text{s.t.} \quad \sum_{i=1}^{n} y_i \alpha_i = 0 \quad \text{and} \quad \forall i \quad \alpha_i \geq 0$$

**Probability Theory**

$$P(A_1, \ldots, A_m \mid B_1, \ldots, B_n) = \frac{P(B_1, \ldots, B_n \mid A_1, \ldots, A_m) P(A_1, \ldots, A_m)}{P(B_1, \ldots, B_n)}$$