**Machine Learning Algorithms**

**Supervised Learning – Task driven – Pre categorized data**

* Linear Regression
  + Find a linear relationship between variables
  + Minimize the sum of squared residuals (the distance between the data points to the function line)
* Logistic Regression
  + Most basic classification algorithm
  + Predict categorical output variables using input variables
  + The outcome is discrete and the function become a Sigmoid curve
  + Probability of a data points falls into certain class given the input variables
* K Nearest Neighbors (KNN)
  + Can be used for both regression and classification
  + Non-parametric algorithm since we are not trying to fit an equation
  + Given a new data point, predict value based on amount k of its nearest neighbors
  + Suitable for complex relationships more than linear
  + K is the hyperparameter
    - Choosing the right K to determine the appropriate fitting
    - Cross validation to find the optimal hyperparameter
* Support Vector Machine (SVM)
  + Originally designed for classification but can also use for regression
  + Drawing decision boundary between data points to separate the data points as well as possible
  + New data points will be classified to where it falls
  + SVM try to find the line that separate the classes with the largest margin possible that would maximize the space between different classes
    - To make the decision boundary generalize well and are robust to noise and outliers
  + Support vectors
    - The data points
    - Sits on the edge of the margin
  + Benefits
    - Very memory efficient
    - Extremely powerful in high dimensions (multiple features)
      * Decision boundary is called Hyperplane then
    - Used kernel functions
      * Identification of highly complex, non-linear decision boundary
      * Ways to turn original features into new, more complex features using kernel tricks -> Implicit feature engineering
* Naïve Bayes
  + Classifier that derived from Bayes Theorem
  + Classify without context and just based on chance of occurrences of a particular thing
    - Email spam and ham
  + Assume the probability of an event happening/thing occurring is independent of each other -> Naïve
  + Very computation efficient and provide a good foundation model
  + Mostly for text-based classification
* Decision Trees
  + Series of questions that would partition the data into several dimensions
  + Create leaf nodes of the tree that are as pure as possible
    - Used Gini Impurity and/or Entropy as threshold to divide
  + Pure
    - Having as least number of data points are misclassified as possible
* Ensemble algorithms
  + Combine many simple models into a more complex model
  + Bagging
    - Train multiple models on multiple different subsets of the data
    - Train the models in parallel
    - Using bootstrapping
    - Random Forests
      * Train multiple trees and vote based on the majority
      * Randomly excluding features from the subsets for different trees in the forest
      * Prevent overfitting and more robust since it eliminates the correlation between the trees
  + Boosting
    - Train the models in sequence
    - Each model focus on fixing the error made by the previous model
    - Combine several weak models to become a strong model (strong learner)
    - Boosted tree
      * Often yield better result than random forest
      * More prone to overfitting
      * Slower to train than random forests (sequential order)
      * Ada Boost
      * Gradient Boosting
      * XG Boost
* Neural Networks
  + Tasks that are too complex for a simple model
    - Classify image to be handwritten numbers from 0 – 9
    - Multiple versions of a handwritten number
    - But all the handwritten version will have commonalities
    - Computers does not know about these complex features but only the pixel intensity of the image
  + Artificial neural network
    - Like using a kernel function with a support vector machine
    - Are designed to implicitly find or create these complex features automatically without guidance from human
    - To do this
      * Adding additional layers of unknown variables between the input and output variables
      * Called single layer perceptron in its simplest form – just a multi features regression tasks
    - Hidden layers
      * Hidden variables in the middle layers represent hidden, unknown features
      * Instead of predicting the output variable directly
      * Would try to predict these hidden variables with the input features first, then try to predict the output target using the new hidden features
        + Several pixels illuminated next to each other can represent a horizontal line, and that would create a new hidden feature
      * One hidden layer would be too simple, so multiple hidden layers to predict hidden layers -> Deep learning
      * Result in complex hidden features

**Unsupervised Learning – Data driven – Unlabeled data**

* Clustering – different from classification
  + Classification is when the classes are known and have training data with true labels available
  + Clustering is when the classes are unknown and want to find unknown clusters from the overall structure of the data and identify potential clusters in the data
  + K-means clustering
    - Most common clustering algorithms
    - K is a hyperparameter – represent the number of clusters
    - Finding suitable K
      * Problems dependent
      * Trial and errors
      * Domain knowledge
    - Randomly selecting centers to for K-clusters and assign all data points to the centers closest to them
    - Then continuously recalculate the cluster centers based on data points assigned to them
    - Until the center of the clusters have stabilized
  + Hierarchical clustering
    - Does not require number of K
    - Continuously divide data into clusters in hierarchical structure
  + DB Scan
    - Find clusters of arbitrary shapes
* Dimensionality reduction
  + Reduce the number of dimensions (features) while keeping as much information as possible
  + Find correlation between existing features and remove potential redundant dimensions without losing information
  + Can also be used as a preprocessing step in supervised learning algorithms
    - Principal Component Analysis (PCA)
      * Strongly correlated features can be combined to create combination feature
        + Combining height and weight features -> shape feature
        + Reduce noise and complexity to the model
        + Keeping as much information as possible
      * Common in large dataset
      * Finding the direction in which most variance is retained – principal component