CS-344 Guide 6 – Machine Learning

- Decision tree learning:
    - Definition: uses a decision tree as a predictive model to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves)
    - Decision tree:
        - A decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility.
        - One way to display an algorithm that only contains conditional control statements.
    - Entropy:
        - Information entropy is the average rate at which information is produced by a stochastic (randomly determined) source of data.
        - The measure of entropy associated with each possible data value is the negative logarithm of the probability mass function for the value.
        -
    - Information gain:
        - Based on the concept of entropy and information content from information theory.
        - Used to decide which feature to split on at each step in building the tree.
        - The amount of information gained about a random variable or signal from observing another random variable.
        - Synonym for Kullback-Leibler divergence.
- Problem Framing:
- Know the following terms:
    - Example:
        - One row of a dataset.
        - Contains one or more features and possibly a label.
    - Label:
        - In supervised learning, the answer or result portion of an example.
    - Features:
        - An input variable used in making predictions.
    - Labeled example:
        - An example that contains features and a label.
        - In supervised training, models learn from labeled examples.
    - Unlabeled example:
        - An example that contains features but no label.
        - Are the input to inference.
        - In semi-supervised/unsupervised learning, are used during training.
    - Model:
        - The representation of what an ML system has learned from the training data.
    - Training data:
        - The actual dataset used to train the model for performing various actions.
    - Training:
        - The process of determining the ideal parameters comprising a model.
    - Bias:

- The systematic error introduced by a sample or reporting procedure.
- An intercept or offset from an origin.
- Stereotyping – can affect collection and interpretation of data, the design of a system, and how users interact with a system.

- Name the basic approaches to machine learning:
  - Supervised learning:
    - Training a model from input data and its corresponding labels.
    - Analogous to a student learning a subject by studying a set of questions and their corresponding answers.
  - Unsupervised learning:
    - Training a model to find patterns in a dataset, typically an unlabeled dataset.
    - The most common use is to cluster data into groups of similar examples.
  - Reinforcement learning:
    - Don't collect examples with labels.
    - Consider teaching a machine to play a basic game and never lose. Setup up a model (often called an agent in RL) with the game, and you tell the model not to get a "game over" screen. During training, the agent receives a reward when it performs this task, which is called a reward function.
- Know the types of ML problems and which ones have proven to be particularly difficult.
  - Classification:
    - Pick one of N labels
    - Example: cat, dog, horse, or bear
  - Regression:
    - Predict numerical values
    - Example: click-through rate
  - Clustering:
    - Group similar examples
    - Example: most relevant documents (unsupervised)
  - Association rule learning:
    - Infer likely association patterns in data
    - Example: if you buy hamburger buns, you're likely to buy hamburgers (unsupervised)
  - Structured output:
    - Create complex output
    - Example: natural language parse trees, image recognition bounding boxes
  - Ranking
    - Identify position on a scale or status
    - Example: search result ranking
  - Difficult ML problems:
    - Clustering – what does each cluster mean in an unsupervised learning problem?
    - Anomaly detection – how do you decide what constitutes an anomaly to get labeled data?
    - Causation – determining if one event or factor causes another is much harder (easy to see that something happened, much harder to understand why it happened)
    - No existing data – if you have no data to train a model, then machine learning can't help you.

- Introduction to Machine Learning:
  - Norvig's reasons for studying machine learning:
    - Reduce the time needed to program.
      - Faster to feed machine learning algorithms examples and train it, then to hand-craft algorithms to do the same.
    - Customize and scale products
      - Better for specific groups of people.
      - Example: localizations of different software.
    - Complete seemingly "unprogrammable" tasks
      - Facial recognization and other tasks we do things subconsciously
    - Changes the way we think of the problem.
      - From mathematical science to a natural science.
      - Experiments and statistics, not logic.
- Framing
  - Be able to give a general explanation of how a model is trained and used.
    - Steps to framing an ML problem:
      - Articulate your problem:
        - Identify which subtype of classification and regression you are using.
        - Flowchart helps assemble the right language to discuss the problem with other ML practitioners.
        - After framing the problem, explain what the model will predict – succinct problem statement.
      - Start simple:
        - Can you simplify the problem?
        - Simplify modeling task – state the problem as a binary classification or uni-dimensional regression problem (or both)
        - Use the simplest model possible – easier to implement and understand.
          - once have full ML pipeline, can iterate on a simple model with more ease.
      - See if any labeled data exists:
        - How much-labeled data do you have?
        - What is the source of your label?
        - Is your label closely connected to the decision you will be making?
      - Design your data for the model:
        - Identify the data that your ML system should use to make predictions (input → output)
        - Each row constitutes one piece of data for which one prediction is made.
        - Only include information that is available at the moment the prediction is made.
      - Determine where data comes from:
        - Assess how much work to develop a data pipeline to construct each column for a row.
        - When does the example output become available for training purposes?
        - Make sure all inputs are available at prediction time in exactly the determined format.

- Determine easily obtained inputs:
  - Pick 1-3 inputs that are easy to obtain and that you believe would produce a reasonable, initial outcome.
  - Which inputs would be useful for implementing heuristic mentioned previously?
  - Focus on inputs that can be obtained from a single system with a simple pipeline.
- Ability to learn:
  - Will the ML model be able to learn?
  - List aspects of your problem that might cause difficulty
- Think about potential biases:
  - Many datasets are biased
  - May adversely affect training and predictions made.

- Compare and contrast regression vs. classification.
  - Classification flow chart:
    - How many categories to pick from?
      - =2 binary classifications (click or no click?)
      - >2 multi-class classifications (the type of animal?)
        - How many categories for a single example?
          - =1 multi-class single-label (which type of animal is this?)
          - >1 multi-class multi-label (what are all the animals in this picture?)
  - Regression flow chart:
    - How many numbers are output?
    - =1 uni-dimensional regression (how many minutes of video will this user watch?)
    - >1 multi-dimensional regression (what is the [latitude, longitude] of the location in the photo?)

- Descending into ML
  - Explain the nature of a linear model.
    - Linear regression is a method for finding the straight line or hyperplane that best fits a set of points.
      - Y = mx + b
      - Y = the value we’re trying to predict
      - M = the slope of the line
      - X = the value of our input feature
      - B = y-intercept.
    - By convention in machine learning, the equation for a model differ slightly:
      - Y’ = b + w1*x1
      - Y’ = the predicted label (the desired output)
      - B = the bias (the y-intercept), sometimes referred to as w0.
      - W1 = the weight of feature 1 (weight is the same concept as the slope “m” in the traditional equation of a line)
      - X1 = a feature (a known input)
    - A more sophisticated model may rely on multiple features, each having a separate weight
      - i.e. y’ = b + w1x1 + w2x2 + w3x3
  - Compare and contrast L2Loss vs. Mean Square Error (MSE):

- Training a model simply means learning (determining) good values for all the weights and the bias from labeled examples.
- Loss: the penalty for a bad prediction.
  - A number indicating how bad the model’s prediction was on a single example.
  - The goal of training a model is to find a set of weights and biases that have low loss, on average, across all examples.
- Squared loss (l2 loss):
  - = the square of the difference between the label and prediction
  - = (observation – prediction(x))^2
  - = (y-y’)^2
- Mean square error (MSE):
  - The average square loss per example over the whole dataset.
  - Sum up all the squared losses for individual examples and then divide by the number of examples:
    - MSE = 1 / N * Riemann Sum (x,y) element of D (y – prediction(x))^2
    - X = the set of features that the model uses to make predictions
    - Y = the label
    - Prediction(x) = a function of the weights and bias in combination with the set of features x.
    - D = a data set containing many labeled examples, which are (x, y) pairs.
    - N = the number of examples in D.

- Reducing Loss:
  - Know the following terms:
  - Hyper-parameter:
    - Are the knobs that programmers tweak in machine learning algorithms.
    - Tuning the learning rate – too small will take too long and too large will cause the next point for the gradient descent on the loss curve to bounce across the bottom of the curve and miss the minimum.
  - Learning rate:
    - Also called the step size.
    - A scalar value that determines the next point in gradient descent algorithms.
  - Explain the nature of (stochastic) gradient descent.
    - Batch = the total number of examples you use to calculate the gradient in a single iteration.
    - Goal: get the right gradient on average for much less computation by choosing examples at random from our data set in order to estimate a big average from a much smaller one.
    - Uses only a single example (batch size of 1) per iteration to calculate the gradient descent.
    - Works with enough iteration but is noisy.
    - Stochastic = indicates that the one example comprising each batch is chosen at random.
- Programming tools:
  - Scikit-learn
  - ^_^