* + In summary, ID3 and C4.5 are entropy-based approaches that are particularly suited for classification tasks. CART is a binary splitting method that can handle both classification and regression tasks, and it's particularly suited for large datasets. CHAID is a statistical-based method that is suited for categorical data. The choice of decision tree algorithm depends on the type of data, the size of the dataset, and the task at hand.
  + If you seen you can classify if not you cant. Nearest neighbor. Doesn’t need exact match. But needs to be close.
  + To calculate P(Y=rich), you need to sum the joint probabilities of Y=rich over all possible values of X1 and X2.
  + Laplace smoothing in naïve bayes classifier is a technique to avoid zero probabilities
  + Gradient Descent can converge into local optimum instead of global ones
  + Complex and Time-Consuming
  + Neural networks are sensitive to the presence of noise
  + Why do we need backpropagation for Artificial Neural Networks (ANN)? Why not just use perceptron learning algorithm? - In ANN, we do not have the true error at each hidden layer - Perceptron assumes availability of ground truth at the output. But in ANN we do not have it for the hidden layers.
  + When a true class label and a predicted class label are the same in the perception algorithm: the weights are not updated
  + A perceptron is a linear classifier.
  + Since activation functions are further combined for hidden layer inputs, the Multilayer Perceptron MLP is a non-linear classifier.
  + the core problem solved in training an Artificial Neural Networks (ANN) is Gradient descent - It is used to search the parameter space to reduce error by changing the weights at a rate proportional to the differential of the weights.
  + Support vectors are the training points closes to the decision boundary.
  + advantages of having decision boundaries with a large margin in Support Vector Machines?
    - A large margin implies there is a high confidence in classification.
    - Decision boundaries with large margin tend to have better generalization errors.
    - If the margin is small, then any slight perturbation to the decision boundary can have quite a significant impact on its classification.
  + the key concepts underlying SVMs are maximizing the margin between the hyperplane and the closest support vectors, using kernel functions to capture complex nonlinear relationships, and the ability to handle high-dimensional data efficiently. These concepts make SVMs a powerful and widely used machine learning algorithm.
  + Examples of data quality problems?: noise outliers, missing values, duplicate data
  + What is noise? Noise refers to the modification of original values
  + What are outliers? Outliers are data oibjects with characteristics that are considerably different than most of the other data objects in the data set
  + How to deal with duplicate data issues? Data cleaning
  + How to handle for missing values? Eliminate data obejcts, estimate missing values, ignore the missing value during analysis
    - * Decision Tree Split:information gain using entropy
      * need to determine what attribute for first split. To do that we need to use a measure of node impurity. Could be gini index or a few others
        + make grid, get entropy for each row, then the total, then the class and then get the information gain

formulas

* P(A|B) = (P(B|A) \* P(A)) / P(B)
  + P(A|B) is the probability of event A occurring given that event B has occurred.
* The equation of the hyperplane can be written as Y = mX + b
* compute the cosine similarity between these 2 vectors d1 and d2:
  + raw: cos(d1, d2) = (d1 d2) / ||d1|| ||d2||
  + excel: =SUMPRODUCT(A1:A3, B1:B3) / (SQRT(SUMSQ(A1:A3)) \* SQRT(SUMSQ(B1:B3)))
* compute the correlation coefficient
  + excel: =CORREL(A1:A4,B1:B4)
* compute the simple matching coefficient
  + =SUMPRODUCT(--(A1:A4=B1:B4)) / (COUNTA(A1:A4))
* compute the Jaccard coefficient
  + don’t need a formula. Literally its just the matching coordinates over the total. So (1,1,0,0) and (0,1,0,1) is .25
* Table

  Description automatically generated
* Gini index
  + Row a gini =1-((B2/D2)^2+(C2/D2)^2)
  + Row b gini =1-((B3/D3)^2+(C3/D3)^2)
  + Weighted gini for A = Row a gini \*D2/D4 + Row b gini \*D3/D4
  + 1st split is best with the lowest weighted gini index
* Entropy
  + entropy(3,7) = -1\*((B2/D2)\*LOG(B2/D2,2)+(C2/D2)\*LOG(C2/D2,2))
  + entropy(5,1) = -1\*((B3/D3)\*LOG(B3/D3,2)+(C3/D3)\*LOG(C3/D3,2))
  + entropy for a = C10\*D2/D4+C11\*D3/D4
* information gain for attribute a = class entropy – entropy for a.
  + note entropy of equal classes ie (8,8) will always be = 1. So no need to calc class
* misclassification error - ex. Node is split 40 yes and 80 no…error is 40/120.
* Euclidean distance
* Manhattan distance |x1 - x2| + |y1 - y2|
* Bayes theorem
* Naïve Bayes Classifier assumes that the attributes are conditionally independent given the class
  + P(C | x1, x2, ..., xn) = P(C) \* P(x1 | C) \* P(x2 | C) \* ... \* P(xn | C) / P(x1, x2, ..., xn) where C is the class variable and x1, x2, ..., xn are the feature values.
  + Based on these information, compute P(Y = rich | X1 = Female, X2 < 40.5) using Naive Bayes assumption of conditional independence.
  + P(Y = rich | X1 = Female, X2 < 40.5) = (P(X1 = Female, X2 < 40.5 | Y = rich) \* P(Y = rich)) / P(X1 = Female, X2 < 40.5)
  + P(Y = rich | X1 = Female, X2 < 40.5) = (0.0362188 \* 0.239181) / 0.1285086
  + P(X1 = Female, X2 < 40.5 | Y = rich) = P(X1 = Female | Y = rich) \* P(X2 < 40.5 | Y = rich)
  + P(X1 = Female, X2 < 40.5 | Y = rich) = (0.0245895 + 0.0116293) \* (0.0245895 + 0.0116293 + 0.253122 + 0.0421768)
  + P(X1 = Female, X2 < 40.5 | Y = rich) = 0.0362188
* forward pass
  + calculate the activation of the hidden layer neurons:
  + z1 = (I1 \* w1) + (I2 \* w2) + b1
  + z2 = (I1 \* w3) + (I2 \* w4) + b1
  + Next, calculate the output of the hidden layer neurons by passing the activation through an activation function, here using sigmoid function:
  + h1 = 1 / (1 + exp(-z1) .. exp(z1) is 2.71828^z1
  + h2 = 1 / (1 + exp(-z2))
  + Then, we compute the activation of the output neuron:
  + z3 = (h1 \* w5) + (h2 \* w6) + b2
  + Finally, we compute the output of the neural network, also using the sigmoid activation function:
  + O = 1 / (1 + exp(-z3))
* determine the attribute that is the first split using the gini index
  + split the dataset by both attributes. the t and f classes are counted by the splits of the class within each.
* accuracy is how accurate the test set is...so take top left and bottom right in a 2x2 grid. cause its TP + TN / all 4 numbers. the flaw in this is its imbalanced. the cancer example...if almost everyone is negative and the model predicts literally everyone, it will have high accuracy but miss the critical step of identifying patients early.
* precision just looks at the ratio of the left column, both rows. TP / (TP + FP)...how precise on the positive class
* sometimes high precision and low recall....so they introduced a measure to capture both and that is F1

|  |  |
| --- | --- |
| TP | FN |
| FP | TN |

* f1 = 2 \* recall \* precision / recall + precision...this is a harmonic average so it actually skews more toward the lower value if there are just 2 values being averaged, unlike traditional average which would skew higher. so you know if f1 is high then both recall and precision are solid
* recall is top row... TP / TP + FN...
* Product A defect rate is 4%, and B is 6%. 60% is A and rest is B. random selected defective product is what % A?
  + P(A | D) = P(D|A) \* P(A) / P(D)
  + P(D|A) = 4% (since product Ahas a defect rate of 4%)
  + P(A) = 60%
  + P(D) = P(D|A) \* P(A) + P(D|B) \* P(B)