Group Project 1: Supervised Learning

Josh Melton and Ivan Benitez

Part 1: Breast Cancer Data

Classifier	F1	Precision	Recall
KNN	0.884	0.920	0.854
Naive Bayes	**0.917**	**0.947**	**0.892**

Table 1. Mean scores from 5-fold cross validation for breast cancer data.

Part 2: Congressional Voting Data

Dataset Version	Classifier	F1	Precision	Recall
Version 1	Decision Tree	**0.949**	0.955	0.944
Version 1	Naive Bayes	0.913	0.886	0.945
Version 2	Decision Tree	0.942	**0.956**	0.928
Version 2	Naive Bayes	0.878	0.847	0.917
Version 3	Decision Tree	0.943	0.937	**0.952**
Version 3	Naive Bayes	0.883	0.852	0.923

Table 2. Mean scores from 5-fold cross validation for congressional voting data.

Part 3: Analysis

1) In Part 1, why do we not use a Decision Tree Classifier?

   A decision tree is built by partitioning the data at each node in the tree using a condition on a given feature. These models work best when partitioning the data at each node leads to a large amount of information gain, which occurs when a partition contains a high proportion of records all with the same class label (ideally all records of one class). With categorical/discrete data a condition can intuitively be determined that breaks the data into distinct groups. With continuous data, a cut-off value must be set in order to create a condition on which to split the data. While decision trees can work with continuous data, in our case malignant tumors can come in all shapes and sizes, so there is not likely to be a cut-off point in each of our features that partitions the data into homogenous subsets. This means that as the tree is built there will not be a large amount of information gain at any given node, so using a decision tree to model this data is not ideal.

2) In Part 2, which missing value strategy seemed to work the best? Why do you think it did?

Across the board, the decision tree models performed better than Naive Bayes. This is because political issues are often highly polarized, so splitting the data based on a vote on a particular issue can lead to a partition of the data with high information gain. Additionally, the Naive Bayes model assumes each feature is independent when calculating probabilities and voting on political issues are not independent events, violating this assumption. Each of the three versions of the decision trees resulted in the best cross validation scores, depending on the metric; though, the difference in scores was generally less than 1%. On the unseen test data, the decision tree model using version two of the data— where NaN values are replaced with a third category— performed the best.

Replacing the missing values with a third category likely resulted in the best model because a congressperson abstaining from voting on certain issues provides additional information that can help determine the congressperson's party. Rather than categorizing voting as just for vs. against, categorizing the data as for vs. against vs. abstain provides more information and likely resulted in partitions with higher information gain, resulting in a better decision tree model.

3) For what kind of datasets do you think a Decision Tree would work best? What about Naive Bayes? KNN?

Decision tree models work by partitioning the data based on a condition for a given feature. This partitioning can work with both categorical and continuous data, but in the case of continuous data, a cut-off/binning of the data must occur to make such conditional branching possible. In these cases, it is possible that the continuous feature does not have such a sharp cut-off, so creating one may not help in distinguishing the class label of each record. These models work best when partitioning the data leads to homogenous subsets which results in high information gain at each node. Decision trees are non-linear, hierarchical models, so they can work well with non-linear datasets that have some semblance of hierarchy among the data features.

Naive Bayes models can also work with categorical (Bernoulli/multinomial) and continuous (Gaussian) data. The Gaussian model works with continuous data and assumes that each feature is normally distributed. While many data in the world are normally distributed, this assumption does not hold for all data, so this model will work best with data that do conform to a normal distribution. Naive Bayes models also assume that all features in the data are conditionally independent, so the model works best on datasets with distinct/independent features. In cases where features may be correlated/interact with one another, the model may be inaccurate.

KNN models, like the others, can work with both categorical and continuous data. This model relies on a notion of distance to classify data points, so it works best with datasets that have a well-defined notion of distance (i.e. can be graphed). This requirement means that categorical data must be transformed to some numeric data in order for a distance function to be applicable. The model can be skewed if certain features of the data are much larger/smaller than others so scaling the data can be helpful in ensuring an accurate distance calculation. KNN is a non-linear classifier, so it can work well with datasets that don’t fit the linear assumptions of other models.