**12.0 - Classification vs. Regression Problems**

In classification problems the response is non-numeric, though it might be coded numerically (e.g. 1 = “Yes”, 0 = “No”). In general the response is either nominal or possibly ordinal, thus the prediction problem is to develop a “model” or “rule” for classifying an what category/level an observation is given a set of predictors (. As was the case with regression problems the predictors can be any mixture of data types (continuous/numeric, ordinal, or nominal) and must dealt with accordingly in developing our predictive model.   
  
For some methods we have covered the extension to classification problems should be clear. For example, the nearest neighbor method covered in Section 11 is easily extended to the classification problem. Rather than take the mean or weighted mean of the response for the nearest neighbors we simply classify an observation to category/level determined by its nearest neighbors. For example, we could simply classify each new/future observation to the class of its nearest neighbor (i.e. ) in the training data. CART and its enhancements (Bagging, Random Forest, Boosting, etc.) also easily extend to classification problems. We build the tree the “usual way” and let *majority rule* in each terminal node. We will have to adjust the “goodness of split” criterion in order to build the tree, but it should be clear that tree-based models are well suited for classification problems.

With the possible exception of nearest neighbor classification and support vector machines (SVM), which we will cover later, the “fitted values” from a classification model are the estimated probability that a -level response is in class given a set of predictor values .

That is our model returns, and we can classify an observation to the class with highest estimated conditional probability.  
  
Because our estimated conditional probabilities need to satisfy the following conditions:  
  
   
  
 2)   
  
we will need to somehow impose these restrictions into the modeling process.

**12.1 – Measuring Prediction Error for Classification Problems**

The basic measures of prediction quality for classification problem are the ***accuracy*** and the ***misclassification rate*** These defined simply as:

Certainly these make sense intuitively, but they are not ONLY measures that can be used. For example if certain misclassifications are more “expensive” to make, then we would like to take that information into account when developing our classification rule. The problems with Accuracy/Misclassification Rate are summarized below.

1. Accuracy/Misclassification Rate assume equal cost for all errors.
2. Is a 99% Accuracy always good? It could excellent, good, mediocre, poor, or flat out terrible depending on the situation. For example, in a binary classification problem (i.e. the response has two levels – “0” and “1”) where 99% of the training data are 0’s and 1% of the training data are 1’s, a rule that says the predicted class for regardless of will be 99% accurate and have a misclassification rate of 1%. What constitutes a good Accuracy depends on the ***Base Accuracy Rate*** which comes from predicting an observation is from the predominant class. In the above example, the Base Accuracy Rate is 99%.
3. The example in (2) above illustrates the need to consider the percent reduction in error or misclassification rate. As an example suppose the Base Accuracy Rate is 80% and hence the Base Misclassification Rate is 20%. Suppose we develop a statistical learning model the increases accuracy from 80% to 90%, thereby reducing the misclassification rate from 20% to 10%. Thus, a statistical learning model reduces error by 50%. Similarly increasing accuracy from 99.9% to 99.99% represents a 90% reduction in error and increasing accuracy from 50% to 80% represents a 60% reduction in error. The idea of percent improvement can be applied to other measures we will consider below as well.

**More Measures of Predictive Performance**

For simplicity we consider the binary classification problem, i.e. the response has two levels (1 = “Yes”, 0 = “No”), when considering prediction accuracy measures in more detail. The accuracy and misclassification rate above can be represented using a ***confusion matrix***.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Predicted | | Row Totals |
| Predicted 1 | Predicted 0 |
| Truth | True 1 | a | b | a+b |
| True 0 | c | d | c+d |
| Total | | a+c | b+d | n = (a+b+c+d) |

Other terminology associated with the confusion matrix,

The ***sensitivity*** is (this is also called the ***Recall*** or True Positive Rate (TPR))

The ***specificity*** of a classification rule is

The ***false positive*** rate (FPR) =

The ***false negative*** rate (FNR) =

What is sometimes of more interest is the predictive value of a classification rule.

The ***positive predictive value (PPV)*** = (this is also called ***Precision***)

The ***negative predictive value (NPV) =***

Usually these come from application of ***Baye’s Rule***:  
  
**PPV**

**NPV**

Calculating these using Baye’s Rule requires that we have some prior knowledge about the probabilities that and in the population of interest, i.e. we need to know or have estimates of and .

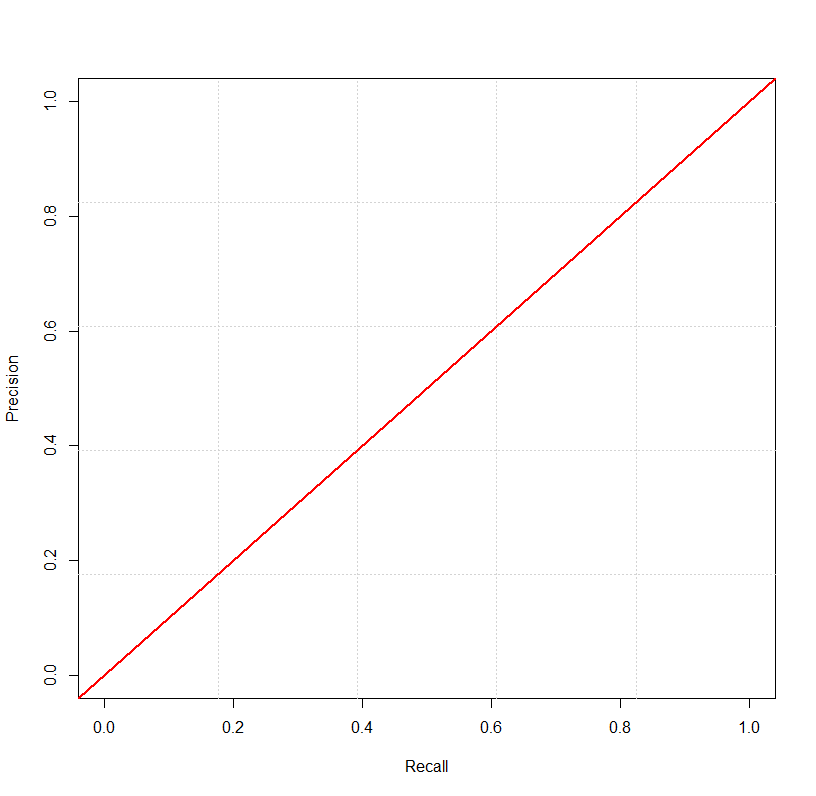
The classification rule for the binary prediction problem will classify an observation given a set of predictor values according to the following rule,

The “logical” choice for , is though in theory we can use any value and there are good reasons for using something other in some cases. As we change the value of the performance of the classification rule change using **any** of the measures above.

Other performance measures that will change with the choice of for a classification rule are the ***F-measure or F1-Score*** and the ***Break Even Point***. The F-measure and Break Even Points are determined by the Precision (PPV) and Recall (Sensitivity or TPR).

and the Break Even Point is the choice of that gives,

The plot below illustrates the Break Even Point.



Break Even Point = .6627

Worse Performance

Better Performance

Remember:

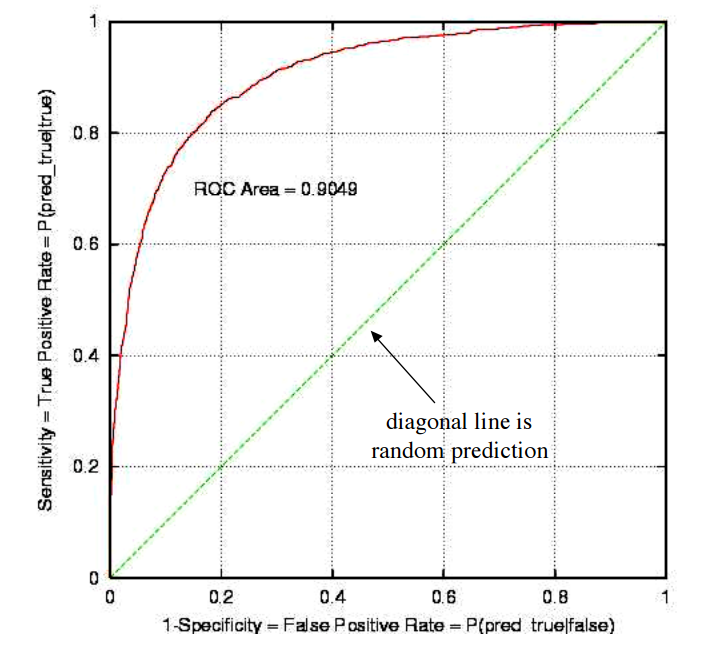
* Recall or TPR or Sensitivity =
* Precision or PPV =

**Receiver Operating Curves (ROC)**

The Receiver Operating Curve (ROC) is a plot of:  
  
 ***Sensitivity*** (Recall) vs. ***1 – Specificity*** (False Positive Rate)

Ideally we would like the sensitivity to be large (near 1) and the 1 – Specificity or the False Positive Rate to be small (near 0). These quantities change as we change the threshold probability in our classification rule which is shown below:

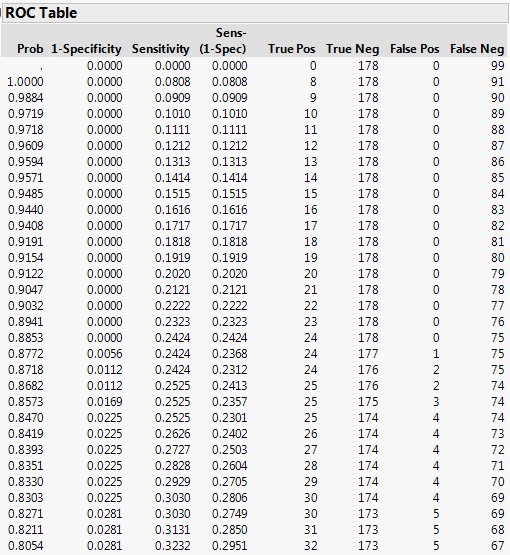
Below is a plot of the ROC for an excellent classification rule for a binary prediction problem. The diagonal line represents the ROC curve for a complete random prediction. The curve represents the predictive performance of a statistical learning model, e.g. from a random forest.



The area beneath the ROC “curve” for a complete random prediction rule is always .50 or 50%. As the predictive performance of classification rule improves the area beneath the ROC curve approaches 1 or 100%. The following adjective scale can be used to discuss the performance of a predictive model in terms the area under the ROC curve, referred to as the AUC (**a**rea **u**nder the **c**urve).

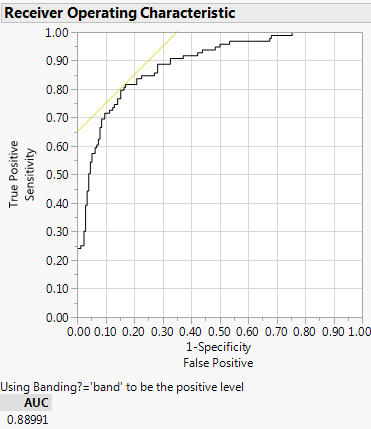
|  |  |
| --- | --- |
| **AUC** | **Predictive Performance** |
| 1.0 or 100% | Perfect Prediction! |
| .90 or 90% | Excellent |
| .80 or 80% | Good |
| .70 or 70% | Mediocre |
| .60 or 60% | Poor |
| .50 or 50% | Random (coin flip) |
| < .50 or 50% | Something is wrong! |

Below is an example of a ROC for a binary prediction problem where we are trying to predict whether or not a printing job will have a problem/defect called “banding”.



The **Prob** column is the table is the cutoff which is used to determine if we predicting banding, i.e.

As decreases sensitivity increases, but so does the false positive rate (FPR) = 1 – specificity.

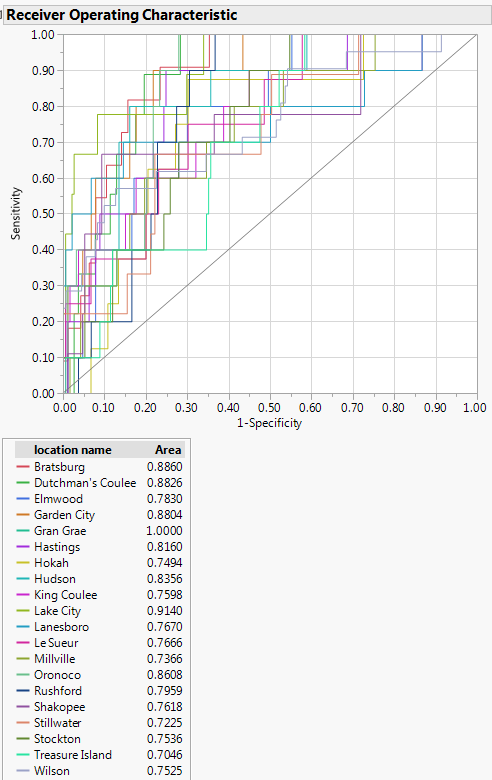


Area Under Curve (AUC) = .8899 or 88.99% nearly excellent prediction!

Small

Large

Good choice for

**Multi-class ROC**Below are the ROC’s for a multi-class prediction problem. Here we are predicting where a soil sample was taken from given the log-concentrations of scandium (Sc), samarium (Sm), uranium (U), and iron (Fe). Some locations are well predicted based on their AUC while others are difficult to classify accurately (e.g. Treasure Island – yes the Casino near Red Wing).  


**Concept of Lift**

Finally we consider another graphical tool for assessing the effectiveness of, or value added by, a statistical learning model for classification problems. We will again focus on the binary prediction problem where the response has two levels.

**Example: Customer Marketing”**Suppose a company wants to do a mail marketing campaign. It costs the company $1 for each mailing. Further suppose they have information on 100,000 potential customers and based on previous mailings they now that approximately 20% of customers positively respond to their mailings. The company has now employed a predictive analyst who knows how to do estimate using data collected from previous mailings. As this information is also available for the 100,000 potential customers being considered for the current mailing the predictive analyst can estimate for these potential customers as well and rank them accordingly.

***Strategy 1***: Mail all 100,000 potential customers

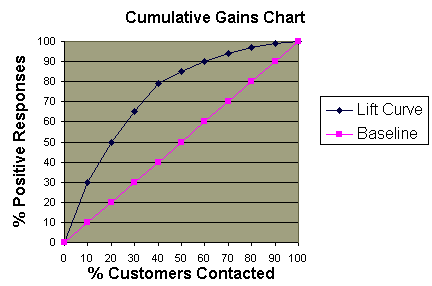
|  |  |  |
| --- | --- | --- |
| Cost  ($) | Total Customers Contacted | Positive Responses |
| 100,000 | 100,000 | 20,000 |

***Strategy 2:*** Use the predictive model to rank customers based upon   
Based upon the model we can estimate the number of positive responses based upon the number of customers contacted ().

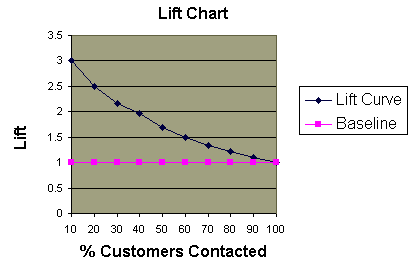
|  |  |  |
| --- | --- | --- |
| Cost ($) | Total Customers Contacted | Expected Positive Responses |
| 10,000 | 10,000 | 6,000 |
| 20,000 | 20,000 | 10,000 |
| 30,000 | 30,000 | 13,000 |
| 40,000 | 40,000 | 15,800 |
| 50,000 | 50,000 | 17,000 |
| 60,000 | 60,000 | 18,000 |
| 70,000 | 70,000 | 18,800 |
| 80,000 | 80,000 | 19,400 |
| 90,000 | 90,000 | 19,800 |
| 100,000 | 100,000 | 20,000 |

Notice the diminishing number of positive responses per 10,000 mailings.

The chart below, called a ***Cumulative Gains Chart***, shows the expected % of positive responses vs. the percentage of the 100,000 customers contacted. The baseline comes from fact that if we mail customers without ranking them we can expect about 10% the 20,000 positive responses per 10% mailed. If we consider the first point on the lift curve we see it has coordinates (10%,30%), which comes from the fact in the first 10,000 **ranked** customers we expect 6,000 positive responses which represents 30% of the 20,000 expected positive responses in the 100,000 potential customers. As the gain diminishes as we move further down our ranked list the lift curve begin taper markedly as the % of customer contacted increases.

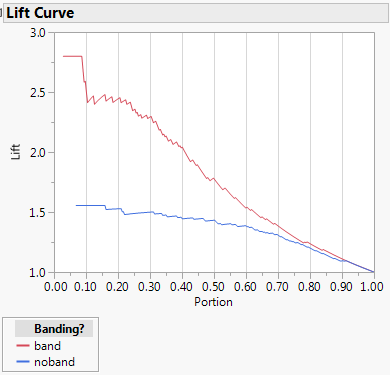


A ***lift chart*** plots the ratio of points on the lift curve relative to the points on the baseline vs. the % of customers contacted.



Again we see the diminishing returns, i..e. decreased lift, as the % of customers contacted increases.

**Example – Lift Curve for the Banding Problem**



By considering only 10% of the ranked process settings based upon our model we will identify about 2.7 times as many scenarios (process settings) where banding is present then if we use no model. Similarly we will identify over 1.5 times as many scenarios where banding is NOT present by using our model.

**12.2 – Dealing with Unbalanced Classes**

For simplicity we will focus on binary classification in our discussion of this issue. It is very common in binary classification problems for the two classes to be highly unbalanced,   
e.g. > 99% 0’s and < 1% 1’s. In such situations simply classifying all cases as 0’s will be highly accurate, thus using this as a measure of predictive performance is a VERY bad idea. In this discussion we will consider some strategies for dealing with highly unbalanced binary classification problems.

**Choice of Metric**

As mentioned above, accuracy is NOT a good measure of predictive performance in situations where classes are highly unbalanced. Below is a non-exhaustive list of metrics that are good to use in highly unbalanced classification problems.

* AUC – the area beneath the ROC curve (pgs. 397-398)
* Precision-Recall Curves (pg. 396)
* F1-statistic (pg. 395)
* Use a cost-based metric, that penalizes models more heavily misclassifications where 1’s are classified as 0’s. This is a setting that can supplied to several of the modeling strategies we will be considering.
* Kappa statistic – compares performance of a classifier to the expected classification accuracy of a purely random classifier. The higher the kappa statistic, the better the classifier.

**Sampling**

In situations where we have very unbalanced classes we can make create balanced classes by:

* Oversampling minority class
* Under-sampling majority class
* Creating more minor class examples (synthetic)

**Choice of Model or Tuning Parameters**

* Some modeling algorithms (e.g. XGBoost) allow for weighting some cases more than others. In highly unbalanced classification problems we can give more weight to minority class cases.
* Some modelling methods are more adept and handling unbalanced problems than others. For example if they allow class weights.
* Use anomaly detection methods (one-class methods) instead.

There are some good discussion on this topic on the internet, here are a couple links worth checking out.

* <https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/>
* <https://www.analyticsvidhya.com/blog/2017/03/imbalanced-classification-problem/>
* <https://www.quora.com/In-classification-how-do-you-handle-an-unbalanced-training-set>