Name: \_\_Sri Vamshi Polela\_\_\_

CSDA 5430 Predictive Analytics

Week 7 Assignment

1. **Explain why predictive accuracy is not the same as goodness-of-fit. How do the strategies, such as holdout, cross validation, and bootstrap sampling, support predictive performance evaluation?**

**Predictive accuracy:** It measures how well a model's predictions match the actual outcomes. (Check Performance on test data).

**Goodness-of-fit:** Another way to see if a model fits well with the given data. (Check performance (Closeness) on trained data).

* Predictive accuracy is more forward-looking, evaluating a model's performance on new data (unseen data), while goodness-of-fit is backward-looking, assessing how well a model describes the data it was trained on.
* Predictive accuracy often involves metrics related to correct predictions, whereas goodness-of-fit may involve metrics related to the closeness of predicted values to actual values in the training data.

**How these Strategies Support predictive performance Evaluation:**

**Holdout Method:**

* Approach: Split the dataset into training and testing sets.
* Purpose: Evaluate the model's performance on unseen data by assessing its accuracy on the testing set.

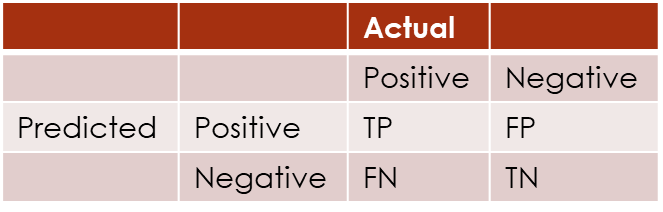
**Cross-Validation:**

* Approach: Divide the dataset into multiple subsets (folds) and iteratively use each fold as a testing set while the others act as the training set.
* Purpose: Provides a more robust estimation of model performance by using different combinations of training and testing sets.

**Bootstrap Sampling:**

* Approach: Randomly sample, with replacement, from the dataset to create multiple bootstrap samples.
* Purpose: Assess the stability and variability of the model's performance by evaluating it on different bootstrap samples.

1. **The Kappa (κ) statistic compares classifier result with ground truth and answers the question of how much better the agreement is (between the ground truth and the machine learning prediction) than would be expected by chance alone. Examine the following confusion matrix for *Quality of Life and Chronic Disease* case study on chapter 13 ppt slide 12 that is generated from C5.0 classification tree model.**

A black text on a white background

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Actual |  |  |
|  |  | severe\_disease | minor\_disease |  |
| Predicted | severe | A=157 | B=72 | A+B=229 |
|  | minor | C=71 | D=143 | C+D=214 |
|  |  | A+C=228 | B+D=215 | N=443 |

* **What is the accuracy of the classifier? How does the accuracy compare to the baseline performance from a ZeroR (zero rule) classifier?**
* Accuracy = = = 0.678 or 67.8%
* As for comparing it to the baseline performance from a ZeroR (zero rule) classifier, the ZeroR classifier simply predicts the most frequent class in the dataset.
* In this case, it would predict the majority class, which is "severe" since A+B >C+D.

So, the accuracy of the ZeroR classifier would be the proportion of the majority class:

* ZeroR Accuracy = = = 0.517 or 51.7%
* Comparing the accuracy of the classifier to the ZeroR classifier, we can see that the classifier performs better than simply predicting the majority class, indicating that it has some predictive power beyond random guessing.
* **Based on the confusion matrix, calculate values for observed agreement, expected agreement, and the kappa score. Comment on the result.**
* Kappa statistics compare classifier result with ground truth.
* 𝑂𝑏𝑠𝑒𝑟𝑣𝑒𝑑𝐴𝑔𝑟𝑒𝑒𝑚𝑒𝑛𝑡 = (𝐴+𝐷) = (‘143+157) = 300
* =
* =
* The Kappa score of 0.35376 suggests a fair to moderate level of agreement beyond what would be expected by chance alone.

**Interpretation of Kappa score:**

* No agreement: ≤ 0
* Poor agreement: 0-0.20
* Fair agreement: 0.20-0.40
* Moderate agreement: 0.40-0.60
* Good agreement: 0.60-0.80
* Very good agreement: 0.80-1

In this case, the Kappa score of 0.35376 falls within the "Fair agreement" range, indicating some degree of agreement between the classifier's results and the ground truth beyond what would be expected by chance alone.

1. **The data file *OwnerExample.csv* contains a small set of predictive model validation results for a classification model on rider mowers, with both actual values (owner or nonowner) and propensities (probability of being owner). In R, load the data and calculate predictive model accuracy, sensitivity and specificity using threshold values of 0, 0.2, 0.3, 0.5, 0.7, 0.8, and 1.0, respectively. You can add more threshold values in between the specified intervals to see changes. Discuss how lowering (or raising) threshold value relates to changes of accuracy, sensitivity, and specificity.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Threshold** | **Accuracy** | **Sensitivity** | **Specificity** | **Interpretation** |
| 0.0 | 0.5 | 1 | 0 | All instances are predicted as positive, resulting in high sensitivity but low specificity. |
| 0.2 | 0.8333 | 1 | 0.6667 | A balance is achieved with higher accuracy, maintaining high sensitivity while improving  specificity. |
| 0.3 | 0.7917 | 0.9167 | 0.6667 | A slight decrease in accuracy with a trade-off between sensitivity and specificity. |
| 0.5 | 0.875 (high) | 0.9167 | 0.8333 | A higher threshold results in increased accuracy and specificity, maintaining good  sensitivity. |
| 0.6 | 0.8333 | 0.8333 | 0.8333 | Balanced performance with equal sensitivity and specificity. |
| 0.7 | 0.7917 | 0.6667 | 0.9167 | A trade-off with higher specificity at the cost of  sensitivity. |
| 0.8 | 0.7917 | 0.5833 | 1 | High specificity is achieved but at the expense of sensitivity. |
| 1.0 | 0.5 | 0 | 1 | All instances are predicted as negative, resulting  in high specificity but low sensitivity. |

Lowering or raising the threshold value in a classification model has a significant impact on accuracy, sensitivity, and specificity.

**Lowering the Threshold:** When the threshold is lowered, more instances are classified as positive, increasing sensitivity but decreasing specificity. This leads to a higher number of true positives (sensitivity) but also more false positives (decreased specificity), resulting in a trade-off between the two metrics. Accuracy may decrease slightly due to the increase in false positives.

**Raising the Threshold:** Conversely, raising the threshold results in fewer instances being classified as positive, leading to higher specificity but lower sensitivity. This means fewer false positives but also fewer true positives, impacting sensitivity. Accuracy may increase due to the reduction in false positives, but it may come at the cost of missing some true positives.

1. Extending above question 3, plot the ROC curve of the predict model to visualize its performance. What is the value for AUC (area under the ROC curve)? Interpret the result.

A graph of a function

Description automatically generated

In this situation, we have Sensitivity and specificity numbers that show how well a model can correctly identify positive and negative cases. The AUC value, which is 0.9375, tells us the model is good at telling apart positive from negative cases.

This value falls within the "Outstanding" range according to the given scale in class. In simpler terms, the high AUC value suggests that the model performs exceptionally well in correctly identifying positive cases while minimizing false positives, making it highly reliable for its intended classification task. Overall, the high AUC suggests the model is doing a good job at its task.

1. Universal Bank is a relatively young bank growing rapidly in terms of overall customer acquisition. The majority of these customers are liability customers (depositors) with varying sizes of relationship with the bank. The customer base of asset customers (borrowers) is quite small, and the bank is interested in expanding this base rapidly to bring in more loan business. In particular, it wants to explore ways of converting its liability customers to personal loan customers (while retaining them as depositors).

A campaign that the bank ran last year for liability customers showed a healthy conversion rate of over 9% success. This has encouraged the retail marketing department to devise smarter campaigns with better target marketing. The goal is to predict whether a new customer will accept a loan offer. This will serve as the basis for the design of a new campaign.

The file *UniversalBank.csv* contains data on 5000 customers. The data include customer demographic information (age, income, etc.), the customer’s relationship with the bank (mortgage, securities account, etc.), and the customer response to the last personal loan campaign (Personal Loan). Among these 5000 customers, only 480 (= 9.6%) accepted the personal loan that was offered to them in the earlier campaign.

In R, perform the following steps as discussed in class on the data. We would like to use various algorisms to make the classification and evaluate performance.

* Step 1: Collecting data (Discuss the business problem and how the data can support the analytics.)

A screenshot of a computer

Description automatically generated

**Business problem:**

* Universal Bank has 5000 customers but most of them are depositors. The bank decided to bring some ways to convert them as personal loan customers (while retaining them as depositors).
* Banks want to increase the personal loan business and the goal is to predict whether a new customer will accept a loan offer or not.

**Data Understanding:**

* **Key variables** include age, income, mortgage, securities account,
* The Important class for analysis is customers who accepted personal loans (Personal.Loan=1). Focusing on this class is pivotal in designing effective campaigns targeted at converting liability customers into personal loan customers. The primary goal is predicting whether a new customer will accept a loan offer, making accurate identification of loan acceptors more crucial than non-acceptors.
* Step 2: Exploring data and preparing data (Remove zipcode, making sure categorical data are factored correctly. Create training partition (60%) and testing partition (40%) with randomized partitioning. Set a seed so the results can be reproduced.)

A screenshot of a computer

Description automatically generated

* Step 3: Training a model on the data (Use the function *rpart()* from the rpart package for the classification tree algorithm. Remember to ignore customer id as part of predictors. Get a plot of the tree.)

A diagram of a graph

Description automatically generated with medium confidence

* Step 4: Evaluating model performance (Evaluate the model performance against the test data with confusion matrix. Identify accuracy, sensitivity, and specificity.)

I took Important class as “Yes”, so that it will calculate according to the customers who accept the loan.

**Accuracy: 0.9855**

**Sensitivity: 0.9479**

**Specificity: 0.9895**

A screenshot of a computer

Description automatically generated

* Step 5: Improving model performance with meta-leaning (Use function *bagging()* in ipred package, function *randomForest()* in randomForest package, and function *boosting()* in adabag package to build models on the training data. You can use all default parameters or a few customized parameters for training. Evaluate model performance the same way as in step 4. Comment your results.)

**Using bagging() function Using randomforest() function**

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**Using boosting() function**

A screenshot of a computer

Description automatically generated

**Comparison of different models:**

| **Model** | **Accuracy** | **Sensitivity** | **Specificity** |
| --- | --- | --- | --- |
| **Decision Tree** | 98.55% | 94.79% | 98.95% |
| **Bagging** | 99% | 92.19% | 99.72% |
| **Random Forest** | 98.75% | 89.06% | 99.78% |
| **AdaBoost** | 99.15% | 93.23% | 99.78% |

* All models exhibit high accuracy, with AdaBoost having the highest sensitivity, making it suitable for scenarios prioritizing correct identification of positive cases. Bagging and Random Forest demonstrate excellent specificity, balancing overall accuracy in identifying both positive and negative cases.
* AdaBoost is the preferred choice due to its high accuracy, coupled with the highest sensitivity, making it particularly well-suited for scenarios where accurately identifying positive cases is of paramount importance.

**Description of Variables in Universal Bank Dataset**

|  |  |
| --- | --- |
| **Variable** | **Description** |
| ID | Customer ID |
| Age | Customer's age in completed years |
| Experience | #years of professional experience |
| Income | Annual income of the customer ($000) |
| ZIPCode | Home Address ZIP code. |
| Family | Family size of the customer |
| CCAvg | Avg. spending on credit cards per month ($000) |
| Education | Education Level. 1: Undergrad; 2: Graduate; 3: Advanced/Professional |
| Mortgage | Value of house mortgage if any. ($000) |
| Personal Loan | Did this customer accept the personal loan offered in the last campaign? |
| Securities Account | Does the customer have a securities account with the bank? |
| CD Account | Does the customer have a certificate of deposit (CD) account with the bank? |
| Online | Does the customer use internet banking facilities? |
| CreditCard | Does the customer use a credit card issued by UniversalBank? |