Classification: Predicting Breast Cancer

Sujay Talanki

8/1/2022

This project uses several different classification methods to predict whether or not a patient has breast cancer.

Loading the dataset:

```r
breast <- read.csv("C:/Users/talan/Downloads/data.csv")
```

I created a generic function below to calculate the K-fold Cross validation accuracy of each model. This model takes in a function (I will create a function for each model) and the value of K (the number of test splits):

```r
CalculateCVAccuracy = function(FUN, K) {
  n = nrow(breast)
  vad.index = sample(rep(1:K,n/K))
  accuracy = 0
  #For loop that computes K-fold CV score
  for (k in 1:K) {
    train = breast[vad.index!=k, ]
    test = breast[vad.index==k, ]
    nk = sum(vad.index==k)
    accuracyk = FUN(train, test)
    accuracy = accuracy + (nk/n) * accuracyk
  }
  #Returns the average K-fold CV score
  return (accuracy)
}
```

- **(a)**: Using **Logistic Regression**:

```r
#Function to simulate logistic regression
logisticRegression = function(train.set, test.set) {
  #Creating the logistic regression model
  logistic.fit = glm(diagnosis ~ ., data = train.set, family = "binomial")

  #Using the training model on the test set
  probability = predict(logistic.fit, test.set, type = "response");
  prediction = rep(0, 1)
  prediction[probability > 0.5] = 1
  test_pred <- ifelse(predict(logistic.fit, newdata = test.set,
                                    type = "response") > 0.5, 1, 0)
  test.accuracy = (mean(test_pred == test.set$diagnosis))*100
  return (test.accuracy)
}
set.seed(1)
CVAccuracy = CalculateCVAccuracy(logisticRegression, 10)
```

The 10-fold Cross-validation accuracy of the logistic regression model is ~ 94.24%.

- **(b)**: Using **Linear Discriminant Analysis (LDA)**:

```
#Function that trains the LDA model
LDA = function(train.set, test.set) {
  lda.fit = lda(diagnosis ~ ., data = train.set)
  pred <- predict(lda.fit, test.set)
  test.accuracy = (mean(pred$class == test.set$diagnosis))*100
  return (test.accuracy)
}
set.seed(2)
CVAccuracy = CalculateCVAccuracy(LDA, 10)
```

The average 10-fold Cross-validation accuracy of the LDA model is ~ 93.93%.

- **(c)**: Using **Quadratic Discriminant Analysis (QDA)**:

```
#Function that trains the QDA model
QDA = function(train.set, test.set) {
  qda.fit = qda(diagnosis ~ ., data = train.set)
  pred <- predict(qda.fit, test.set)
  test.accuracy = (mean(pred$class == test.set$diagnosis))*100
  return (test.accuracy)
}
set.seed(3)
CVAccuracy = CalculateCVAccuracy(QDA, 10)
```

The average 10-fold Cross-validation accuracy of the QDA model is ~ 94.28%.

This next part uses the Support Vector Machine (SVM) classifier. As it is more algorithmically intensive, I will compute a 2-fold Cross Validation accuracy for each classifier.

- **(d)**: Using a **Support Vector Machine (SVM)**. This will be a **linear** SVM and I will tune the model in order to find the best cost.

```
#Function that trains the SVM linear model
SVMlinear = function(train.set, test.set) {
  #Finding the optimal cost
  tune.fit = tune(svm, diagnosis ~ ., data = train.set,
  kernel = "linear", type = 'C-classification', ranges = list(cost =
  seq(0.01, 10, 0.5)))
  best.fit = tune.fit$best.model

  #Finding test accuracy
  test_pred = predict(best.fit, test.set)
```

```r
  test.accuracy = (mean(test_pred == test.set$diagnosis))*100
  return (test.accuracy)
}
set.seed(4)
CVAccuracy = CalculateCVAccuracy(SVMlinear, 10)
```

The average 10-fold Cross-validation accuracy of the SVM linear model is ~ 95.99%.

- **(e)**: Using a **Support Vector Machine (SVM)**. This will be a **radial** SVM and I will tune the model in order to find the best cost.

```r
#Function that trains the SVM radial model
SVMRadial = function(train.set, test.set) {
  #Finding the optimal cost
  tune.fit = tune(svm, diagnosis ~ ., data = train.set,
  kernel = "radial", type = 'C-classification',
  ranges = list(cost = seq(0.01, 10, 0.5)))
  best.fit = tune.fit$best.model

  #Finding test accuracy
  test_pred = predict(best.fit, test.set)
  test.accuracy = (mean(test_pred == test.set$diagnosis))*100
  return (test.accuracy)
}
set.seed(5)
CVAccuracy = CalculateCVAccuracy(SVMRadial, 10)
```

The average 10-fold Cross-validation accuracy of the SVM radial model is ~ 96.15%.

- **(f)**: Using a **Support Vector Machine (SVM)**. This will be a **polynomial** SVM and I will tune the model in order to find the best cost.

```r
#Function that trains the SVM polynomial model
SVMPolynomial = function(train.set, test.set) {
  #Finding the optimal cost
  tune.fit = tune(svm, diagnosis ~ ., data = train.set,
  kernel = "polynomial", type = 'C-classification',
  ranges = list(cost = seq(0.01, 10, 0.5)))
  best.fit = tune.fit$best.model

  #Finding test accuracy
  test_pred = predict(best.fit, test.set)
  test.accuracy = (mean(test_pred == test.set$diagnosis))*100
  return (test.accuracy)
}
set.seed(6)
CVAccuracy = CalculateCVAccuracy(SVMPolynomial, 10)
```

The average 10-fold Cross-validation accuracy of the SVM polynomial model is ~ 94.62%.

Conclusion: Most of the classification methods result in a CV test accuracy of around 94% - 96%. It appears that the SVM linear and SVM radial models predict heart disease the best, but all classifiers generally work equally as well.