CAB320 - Artificial Intelligence
Assignment 2: Machine Learning

Vanessa Gutierrez (n9890394), Glenn Christensen (n9884050), Marius Imingen (n9884076)
June 3, 2017

1 Methodology

Using numpy functions we split the file into two arrays, X and Y. We excluded the classification and ID-number data from X since the number is arbitrarily assigned and we don't want any of our classifiers to use any correlation between the ID-number and whether a tumor is malignant or benign. The diagnosis was also excluded from the X-list because we want to give the model the classification in the Y data set.

To split our data into training and testing, we implemented a partition_dataset function. This method randomizes the data and selects and saves 80% for training and 20% for testing, and returns appropriate arrays for X and Y training and validation data sets.

We decided to implement k-fold cross-validation. Since k-fold allows each training data point to be used for validation once, and for validation k-1 times, with lower variance of the resulting estimate, we figured this method allowed us to use our limited data most efficiently to create the best model[1]. We used 10 folds to compromise between the bias-variance tradeoffs of lower or higher k values[2]. Using k-fold validation, we were able to select the classifier that predicted the classifications of validation data most accurately.

To determine the best number of neighbors to use for the Nearest Neighbors classifier, we used k-fold cross-validation on each classifier created with a different number of neighbors ranging from 1 to 15, then chose the classifier with the highest accuracy (Figure 2). We ensured that K always was an odd number so that there could never be a tie of classifications between the neighbours.

To determine which type of kernel to use for the Support Vector Machine classifier, we followed a similar method: we used k-fold cross validation on classifiers created with linear, RBF, Sigmoid and polynomial kernels. The k-fold validation and predictions for the classifier created with a polynomial kernel were exceptionally slow and could not finish within a reasonable amount of time, and we ultimately decided to remove the polynomial kernel from the decision process. We selected the most accurate classifier of those created with the other three kernel types (Figure 3).We can infer our data has a linear-like separation based off of this kernel's significantly higher accuracy.

---

[1] Schneider, Jeff. "Cross Validation." Cross Validation. February 1997. Accessed June 03, 2017. https://www.cs.cmu.edu/~schneide/tut5/node42.html.
[2] Stanford University Statistics Lecture Slides, 15. Accessed June 03, 2017. lagunita.stanford.edu/c4x/HumanitiesScience/StatLearning/asset/cv_boot.pdf

2 Classifier Performance

The Naive-Bayes classifier had 100% prediction accuracy on the validation data and 93.89% prediction accuracy, a 6.14% error, on the testing data.

The Nearest Neighbor classifier (using a value of 7 nearest neighbors) had 93.64% prediction accuracy on the validation data and 92.11% prediction accuracy, a 7.89% error, on the testing data. Our Nearest Neighbors classifier used 5 neighbours, and had the highest validation data classification accuracy out of the classifiers with other neighbor values, at 97.83% (Figure 1).

The Decision Tree classifier had 95.65% prediction accuracy on the validation data and 93.85%, a 6.14% error, prediction accuracy on the testing data.

Running predictions with the Support Vector Machine classifier(s) took considerably longer (minutes) than the other three classifiers (a second or less). The Support Vector Machine classifier had 97.83% prediction accuracy on the validation data and 92.98% prediction accuracy, a 7.02% error, on the testing data.

3 Summary of Results

Overall, the four classifiers had accuracy and error scores within .01, or 1% of each other. Naive Bayes was the most accurate and had the lowest error by a .01% margin over the Decision Tree classifier. The Support Vector Machine classifier's accuracy was very closely trailed by the Nearest Neighbors' less than 1% worse.

Figure 1: Classifiers' Accuracy and Error Scores for
Predicting Training, Validation, and Testing Data Classifications

Classifiers	Training Data Prediction		Validation Data Prediction		Testing Data Prediction	
	Accuracy Score	Error (1 - Accuracy)	Accuracy Score	Error (1 - Accuracy)	Accuracy Score	Error (1 - Accuracy)
Naive Bayes	0.9389	0.0611	1.0	0	0.9389	0.0614
Nearest Neighbours (n_neighbours = 7)	0.9364	0.0636	1.0	0	0.9211	0.0789
Decision Tree	1.0	0	0.9565	0.0435	0.9386	0.0614
Support Vector Machine (kernel = linear)	0.9608	0.0392	0.9782	0.0218	0.9298	0.0702

Figure 2: Best Accuracy of Validation Data Predictions in Cross Validation for Nearest Neighbors Classifiers Using K Number of Neighbors

K	Accuracy Score	Error (1 - Accuracy)
1	0.9556	0.0444
3	0.9565	0.0435
5	0.9783	0.0217
7	1.0	0
9	1.0	0
11	1.0	0
13	1.0	0
15	1.0	0

Figure 3: Best Accuracy of Validation Data Predictions in Cross Validation for Support Vector Machine Classifiers Using Different Kernels

Kernel Type	Accuracy Score	Error (1 - Accuracy
linear	0.9782	0.0218
RBF	0.7391	0.2609
Sigmoid	0.7391	0.2609