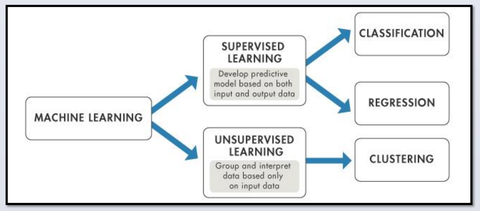
Diagnosing Breast Cancer through Machine Learning

Abstract

Currently, the most common diagnostic method for breast cancer is a mammogram, essentially an x-ray of compacted breast mass, which yields an 87% successful diagnostic rate. This focus of this study was utilizing machine learning classifiers in order to develop a new method for the diagnosis of malignant breast tumors. An initial test of a variety of linear and nonlinear classifiers trained on the UCI "Breast Cancer Wisconsin (Diagnostic)" Data Set, and tested using 10-fold cross validation, was used to identify linear discriminant analysis as the optimal classifier for the diagnostic model. It was then retrained, and applied to new data (a dedicated 20% of the dataset mentioned above). The final product was a complete diagnostic system which produced a 92.11% successful diagnostic rate, over 5% greater than mammograms, leading to the acceptance of the hypothesis, and the rejection of the null. While the results indicate a working model for the diagnosis of a malignant breast tumor, they also reveal important information about data from fine needle aspirates, such as their tendency to be split linearly. In addition, this model acts as a proof of concept for the application of machine learning and other forms of unsupervised computation for the diagnosis of diseases.

Predictive Machine Learning



Predictive modeling is a subset of machine learning, which focuses on building a model that is capable of making predictions. Typically, such a model includes a machine learning algorithm that learns certain properties from a training dataset in order to make those predictions. Predictive modeling can be divided further into two sub areas: Regression and pattern classification. Regression models are based on the analysis of relationships between variables and trends in order to make predictions about continuous variables. In contrast to regression models, the task of pattern classification is to assign discrete class labels to particular observations as outcomes of a prediction.

In relation to the focus of this study, six classifiers were tested, including a range of linear and nonlinear classifiers, listed below:

1. Linear
   a. Logistic Regression (LR)
   b. Linear Discriminant Analysis (LDA)
2. Nonlinear
   a. K-Nearest Neighbors (KNN)
   b. Classification and Regression Trees (CART)
   c. Gaussian Naïve Bayes (NB)
   d. Support Vector Machines (SVM)

Problem and Hypothesis

Problem: Can a machine learning based predictive model be used to diagnose a malignant breast tumor? Which predictive classifier has the highest successful diagnostic rate? How does a machine learning approach compare to the diagnostic rate of mammograms?

Most literature indicates that predictive models can be used to differentiate between two classifications, based on a set of input features. Initial statistical analysis of the dataset indicates a wide distribution of physical properties and values, with predetermined groups, which should in theory correlate with a successful linear or nonlinear distinction between malignant and benign tumors.

Hypothesis: A trained predictive machine learning model will prove more effective in diagnosing a malignant tumor than other diagnostic procedures, most notably mammograms.

Null: Machine Learning cannot be applied to the diagnosis of breast cancer, represented by a success rate of 50% ± 10%

Implementation

```
def main():
    #Splits the dataset 80 / 20
    array = dataset.values
    features_mean= list(simplifiedDataSet.columns[0:10])

    X = dataset.loc[:,features_mean]
    Y = dataset.loc[:, 'Diagnosis']

    X = X.astype('long')
    Y = Y.astype('long')

    validation_size = 0.20
    seed = 7
    X_train, X_validation, Y_train, Y_validation =
model_selection.train_test_split(X, Y, test_size = validation_size,
random_state = seed)
```

```
# evaluate each model in turn
results = []
names = []
for name, model in models:
  kfold = model_selection.KFold(n_splits=10, random_state=seed)
  cv_results = model_selection.cross_val_score(model, X_train, Y_train, cv=kfold, scoring=scoring)
  results.append(cv_results)
  names.append(name)
  output = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
  print(output)
```

```
# Make predictions on validation dataset
lda = LinearDiscriminantAnalysis()
lda.fit(X_train, Y_train)
predictions = lda.predict(X_validation)
print(accuracy_score(Y_validation, predictions))
print(confusion_matrix(Y_validation, predictions))
print(classification_report(Y_validation, predictions))
```

Code Analysis

Section 1:

The first section of code contains standard procedures for the pre-learning period, including dedicating sections of the dataset to training / testing, defining which columns of the *dataFrame* contain the data, and which columns contain the result (diagnosis). In addition, once every block of necessary information is obtained, variables *X_train*, *X_validation*, *Y_train*, and *Y_validation*, are defined according to their respective values.

Section 2:

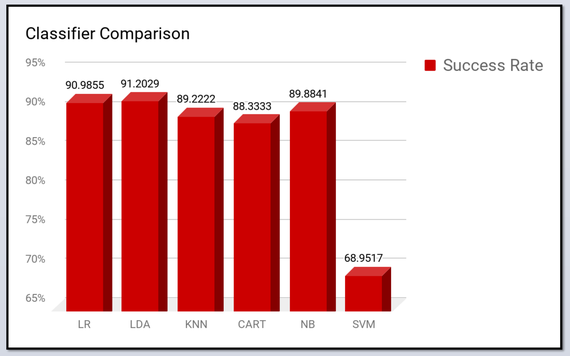
The second section of code displayed assumes an array of classifiers, which was defined but not shown, in order to perform a *for* loop, and execute the same procedure for each classifier. The procedure focuses on actually training the model, by referencing SciKit Learn (a Python library which contains the math behind the various classifiers which I utilized), and feeding in the variables defined in the first section. In addition, this section contains the implementation for 10-Fold cross validation, a method of intermittent testing during the training phase, the output of which can be seen in *Figure 1*.

Section 3:

Utilizing the data from the previous section, specifically which classifiers performed best, the final section of code applies the trained classifier to the remaining 20% of data, in order to simulate the accuracy of the model in a real-world environment.

Results

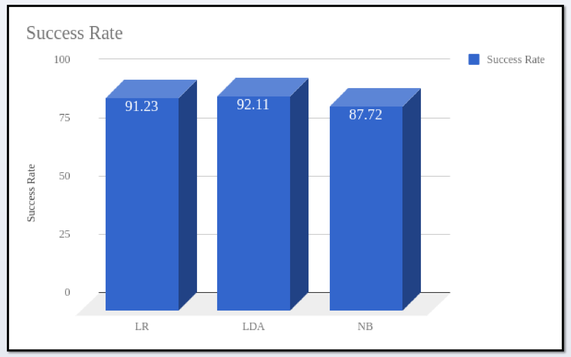
Figure 1: Results of 10-Fold Cross Validation on 6 Classifiers



	Success Rate
Logistic Regression	90.9855
Linear Discriminant Analysis	91.2029
K-Nearest Neighbors	89.2222
Classification and Regression Trees	88.3333
Naive Bayes	89.8841
Support Vector Machines	68.9517

Results Cont.

Figure 2: Results of Testing on New Data



	Success Rate
Logistic Regression	91.23
Linear Discriminant Analysis	92.11
Naive Bayes	87.72

Summary of Data & Conclusion

The results of 10-Fold Cross Validation, *Figure 1*, indicated that Linear Discriminant Analysis, Logistic Regression, and Gaussian Naïve Bayes had the highest initial success rates, producing a correct diagnostic rate of 91.2, 90.99, and 89.88 percent respectively. During the final stage (where the classifiers above were retrained and tested on new data, *Figure 2*), Linear Discriminant Analysis proved to be the most effective classifier for the given model.

In summary of this study, the hypothesis is accepted: A machine learning approach to breast cancer diagnosis performs better (yields a higher success rate) than mammograms. More specifically, the final trained model utilizing the Linear Discriminant Analysis classifier was able to correctly differentiate between malignant and benign diagnoses with a 92.11% success rate. In addition, linear classifiers produced a higher successful diagnostic rate than nonlinear, indicating that physical data from fine needle aspirates should be approached with linear classifiers.

Bibliography

"The Python Language Reference." The Python Language Reference Python 3.6.0 Documentation. The Python Software Foundation, n.d. Web. 26 Dec. 2016.

Brownlee, Jason, et al. "Machine Learning Mastery." Machine Learning Mastery, 17 Jan. 2018, machinelearningmastery.com/.

"Breast Cancer." Mayo Clinic, Mayo Foundation for Medical Education and Research, 17 Jan. 2018, www.mayoclinic.org/diseases-conditions/breast-cancer/symptoms-causes/syc-20352470.

Walia, Anish Singh. "Types of Optimization Algorithms Used in Neural Networks and Ways to Optimize Gradient Descent." Towards Data Science, Towards Data Science, 10 June 2017,

"Breast Cancer Wisconsin (Diagnostic) Data Set." UCI Machine Learning Repository , archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic).