Cancer diagnosis using machine learning

A digital image of a person's body

Description automatically generated

Name: Kyle-Anthony Hay

Class: CSCI-49376 – Big Data

Professor: You Wu

**Random Forest Algorithm(96%)**

Justification:

The first algorithmic approach in this project is the **Random Forest** Algorithm. This was chosen after observing the high number of features in the dataset. It is possible that some features are irrelevant or might result in misleading results. Since this algorithm takes the amalgamation of several decision trees from random samples of the dataset, the effect of potential ambiguous values is reduced. That is, it is more robust concerning noise in the dataset. It is also optimal for **Non-Linear Relationships**, which is advantageous since the features might not directly or proportionally increase/decrease in relation to the benign (B) or malignant (M) classifications.

Procedure:

1. First, the diagnosis was represented by integers for the algorithm to utilize. It was saved in a new column “indexeddiagnosis.” This operation was saved in a variable called “labelIndexer”
2. The columns of the dataset were assembled into a column vector called ”features” after removing the old column “diagnosis” This operation was saved in a variable called “assembler”
3. A RandomForestClassifier was instantiated with the parameters “indexedDiagnosis” and “features”. This governs how the algorithm will be applied to the dataset. This operation was saved in a variable called “rf”
4. The pipeline(the order of operations) was initialized as: (labelIndexer, assembler, rf)
5. The model was then trained on the training data according to the pipeline’s specifications. The Model was stored in a variable “model”
6. The test data was then transformed according to the pipeline and the model was used to predict the results.
7. The model was then evaluated according to the “indexedDiagnosis” and “prediction” fields, and the accuracy, F1, Precision, Recall, and ROC was measured

Results:

Test Accuracy = 0.978142

F1 Score = 0.978142

Precision = 0.978142

Recall = 0.978142

ROC AUC = 0.997928

A graph of a function

Description automatically generated

**Logistic Regression Algorithm (94%)**

This algorithm was chosen as it was specifically designed to produce binary classifications which was optimal for this dataset. It is also simple in design, resulting in it being easily understood. Its simplicity also results in faster computational time.

Procedure: (similar to Random Forrest)

1. First, the diagnosis was represented by integers for the algorithm to utilize. It was saved in a new column “indexedDiagnosis.” This operation was saved in a variable called “labelIndexer”
2. The columns of the dataset were assembled into a column vector called ”features” after removing the old column “diagnosis” This operation was saved in a variable called “assembler”
3. A LogisticRegression was instantiated with the parameters “indexDiagnosis” and “features”. This governs how the algorithm will be applied to the dataset. This operation was saved in a variable called “lr”
4. The pipeline(the order of operations) was initialized as: (labelIndexer, assembler, lr)
5. The model was then trained on the training data according to the pipeline’s specifications. The Model was stored in a variable “lrModel”
6. The test data was then transformed according to the pipeline and the model was used to predict the results.
7. The model was then evaluated according to the “indexedDiagnosis” and “prediction” fields, and the accuracy, F1, Precision, Recall, and ROC was measured

Results:

Test Accuracy = 0.958824

F1 Score = 0.9589

Precision = 0.959047

Recall = 0.958824

ROC AUC = 0.992879

A graph of a function

Description automatically generated

Comparison of two algorithms:

Both algorithms performed well. However, the Random Forest algorithm had a slight edge of 2% over the Logistical Regression algorithm in accuracy. This may be a result of the algorithm's poor predicting capabilities on data with relatively high variation. However, I believe that the sample data’s variability was low resulting in a slight but noticeable drop in performance.