**Breast Cancer Data Report**

In the spike of human error in the health care industry, machine learning solutions have been considered as a great alleviating tool helping doctor verify conclusions. One can think of this as a third eye. Amid that vision, we selected the breast cancer dataset to explore possible models to classify if a given tumor is benign or malignant. This dataset, originally from the University of Wisconsin, consists of 30 features and 569 instances. The attributes set contains the radius, texture, perimeter, area, and more about a given tumor. It is also worth noting that this data has been classified and we will make our training based on these classifications.

Regularly, when you are given any dataset one must understand it. We used a series of plots to visualize and understand the data better. In addition, we used an array of methods from machine learning and deep learning to come out with possible solutions.

We used a heat map to observe the correlation between features. We discovered some both low and high correlation between features, as expected, yet some combination of features were very interesting. For example, mean compactness and mean concave points. In addition to the heat map, we used a paired plot to visualize a feature-by-feature plot, two features at the time. We assign different colors to differentiate benign vs malignant.

In addition to visualization, we used a scaler to scale the data from the data set. The main reason for this is that most machine and deep learning algorithm computes the Euclidian distance between data points and those features with a high magnitude will weight higher relative to others features with low magnitude. The Standard Scaler from Scikit Learn, uses a straightforward formula, z = (x - u) / s, where x is the value to be scale, u is the mean of the samples, and s is the standard deviation. This is great because as shown in class, models can produce better results after scaling.

To measure the performance of our models, which we will expand below, we split our dataset into two sets: training and test set. We set the test size to be 30% overall and they will be picked at random.

Although not mentioned in class, we started exploring by using a simple method, a Decision Tree Classifier. In nutshell, "leaves represent a class label and branches represent conjunctions of features that lead to those labels"[1]. We were surprised by the outcome of this model. Although not perfect, this model scored a precision, a recall, and an f1-score of 93% percent. Another method that we applied was a Logistic Regression model, which showed much more improvement. This model scored a precision, a recall, and an f1-score of 98% which is very satisfying. Yet because we are talking about cancer, we were eager to explore other methods we learned in class.

Searching for better answers, we used a Support Vector Machine model in conjunction with a Grid Search that helped us tune the hyper-parameters of the Support Vector Machine. We provided a set of parameters to the Grid Search object to determine the best estimator. Our results concluded that best estimator is one with a C value of 10000, a gamma value of 1e-4, and a radial basis kernel function. Surprisingly, our report showed the worst result compared to the Logistic Regression model scoring a 95%. Because of that, we decided to add one more value for gamma and C. We found a much better result with a gamma value of 1e-5, scoring 98% for all, precision, recall, and f1-score.

By not been able to do better than Logistic Regression, we decided to use a Multi-Layer Perceptron Classifier (Neural Network). We played around with different hidden layers configuration and came out with better results. Using a hidden layer setup of (20, 10, 30), our neural network was able to score 99% for precision, recall, and f1-score.

In overall, this was pretty good. One can argue that this is still not perfect for health care, yet in my opinion, these models can't really be used to fully diagnose cancer in patients without human supervision. In other to improve these models one can keep exploring the data and try to find that don't have a high correlation. Given more time, one can find a method to reduce the dimensionality of the data, i.e., removing some features that don't have much impact on the classification problem.

Reference:

[1] https://en.wikipedia.org/wiki/Decision\_tree\_learning