Applying Machine Learning (ML) bioinformatic best practices to the publicly available Wisconsin Diagnostic Breast Cancer (WDBC) [dataset](<https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29>) to make a diagnostic for breast cancer in Python.

1. Introduction:

Background on WDBC dataset

The WDBC dataset created by [Dr. Wolberg](# 2), et al. contains 30 features computed for 569 unique instances from a digitized image of a fine needle aspirate (FNA) of a breast mass. The total attributes totals at 32 because of the unique instance’s ID and the diagnosis. They describe the characteristics of the cell nuclei present in the image.

Here is a quick summary of the attributes as described at the link above:

1) ID number 2) Diagnosis (M = malignant, B = benign) 3-32)

Ten real-valued features are computed for each cell nucleus:

* a) radius (mean of distances from center to points on the perimeter)
* b) texture (standard deviation of gray-scale values)
* c) perimeter
* d) area
* e) smoothness (local variation in radius lengths)
* f) compactness (perimeter^2 / area - 1.0)
* g) concavity (severity of concave portions of the contour)
* h) concave points (number of concave portions of the contour)
* i) symmetry
* j) fractal dimension ("coastline approximation" - 1)

The mean, standard error and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius.

All feature values are recoded with four significant digits.

Missing attribute values: none

Class distribution: 357 benign, 212 malignant

Background on applications to WDBC dataset

So people have been tackling this problem for a long time now since the data was donated in 1995. Heck, if you just go to the site in the link, there is ~ 7 relevant papers on it. Furthermore, I’ve had college professors use versions of this data for homework in class with ML. In fact, [Kaggle]( https://www.kaggle.com/uciml/breast-cancer-wisconsin-data) also has this on their site for people to tackle for ML. It really is a great data science problem to work on because the relevance is significantly life saving!

So what is new today? And how might we apply these recent discoveries to our old problems? Well…

A recent publication by [Randal S. Olson](http://www.randalolson.com), et al. in 2017 provides insightful best practice advice for solving bioinformatic problems with machine learning, [“Data-driven Advice for Applying Machine Learning to Bioinformatics Problems”](<https://arxiv.org/abs/1708.05070>).

Looking at the abstract briefly, they analyzed “13 state-of-the-art, commonly used machine learning algorithms on a set of 165 publicly available classification problems in order to provide data-driven algorithm recommendations to current researchers.” Wow! That is very nice of them. But wait! Even better is that one of those 165 publicly available classification problems comes from this WDBC study (see Table 2 in their paper).

From their research, they were able to provide a “recommendation of five algorithms with hyperparameters that maximize classifier performance across the tested problems, as well as general guidelines for applying machine learning to supervised classification problems.” These recommendations are summarized below as follows:

A screenshot of a cell phone

Description automatically generated

Therefore, I thought it would be a great learning experience for me as an ambitious and aspiring data scientist to code these same 5 recommendations with the same [scikit-learn]( https://scikit-learn.org/stable/user\_guide.html#user-guide) implementations for tuning the hyperparameters.

Because lets be honest! When it comes to diagnosing breast cancer, we want to make sure we don't have too many false-positives (you don't have cancer, but told you do and go on treatment) or false-negatives (you have cancer, but told you don't and don't get treatment). I like to think of this problem as, every bit of accuracy gained is potentially saving a life!

My approach:

The Data was split into 80% training (~455 people) and 20% testing (~114 people).

Using the 80% training data, several different models were evaluated through k-fold Cross-Validation set GridSearchCV, which iterates on different algorithm's hyperparameters:

* Logistic Regression
* Support Vector Machine
* Neural Network
* Random Forest
* Gradient Boost
* eXtreme Gradient Boost

All of the models performed well after fine tuning their hyperparameters, but the best model is the one the highest overall accuracy. Out of the 20% of data withheld in this test (114 random individuals), only a handful were misdiagnosed. No model is perfect, but I am happy about how accurate my model is here. If on average less than a handful of people out of 114 are misdiagnosed, that is a good start for making a model. Furthermore, the Feature Importance plots show that the "concave points worst" and "concave points mean" were the significant features. Therefore, I recommend the concave point features should be extracted from each future biopsy as a strong predictor for diagnosing breast cancer.

Using the Wisconsin Breast Cancer dataset, several different Machine Learning models were created and evaluated with k-fold Cross-Validation using GridSearchCV, which also iterates on different algorithm's hyperparameters:

* Logistic Regression
* Support Vector Machine
* Neural Network
* Random Forest
* Gradient Boost
* eXtreme Gradient Boost

References.

1. <https://arxiv.org/abs/1708.05070v2>
2. Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

t

t

### Looking at the Heatmap of the Pearson Correlation Coefficient Matrix for Features

#### Figure 1

The Heatmap shown in Figure 1 gave us insight into how well correlated our features were with respect to each other and the diagnosis. If you look at the far left column, you can see that most of the features are well correlated with the diagnosis row. The strongest correlated feature to diagnosis is the `concavity points\_worst`, with a Pearson Correlation Coefficient to `diagnosis` of ~0.79. Hence, why the Machine Learning (ML) models performed so well as described below.

### Looking at the Confusion Matrix Plots

#### Figures 2.A, 3.A, 4.A, 5.A, 6.A, 7.A, 8.A, 9.A, 10.A, 11.A, 12.A, and 13.A

When it comes to diagnosing breast cancer, we want to make sure we don't have too many false-positives (you don't have cancer, but told you do and go on treatment) or false-negatives (you have cancer, but told you don't and don't get treatment). Therefore, the highest overall accuracy model is chosen (the accuracy is the sum of the diagonals on the confusion matrix divided by the total). You can see that all of the models had less than a handful of false-positives and false-negatives, meaning they were all extremely accurate since there was 114 observations for the test set.

### Looking at the Variable Importance Plots

#### Figures 2.B, 3.B, 4.B, 5.B, 8.B, and 9.B

Most of the `ensemble` models have a parameter called `predict\_proba\_`, which allows the output of the most significant features of the model based on their probability through majority vote via either the `gini` index or `entropy`. The top 5 variables that appear most important for helping the model make a prediction are:

\* `concave points\_worst`

\* `concave points\_mean`

\* `radius\_worst`

\* `area\_worst`

\* `perimeter\_worst`

Not a surprise that these same variables also show a high correlation with diagnosis in the Heatmap from Figure 1. Thus, it makes sense why these variables turned out so important.

### Looking at Area Under the Curves (AUC) Plot

#### Figure 14

All of the models had `Excellent` Area Under the Curves (AUC) because their values were greater than 90% (really close to 99% for all), meaning they all would serve as excellent diagnostics. This is also shown by their high Specificity and Sensitivity values.

### Reflecting on the Recommended Hyperparameters by Olson

All of the models performed well after fine tuning their hyperparameters, but the best model is the one with the highest overall accuracy.

In this analysis, using Olson's recommendations for Support Vector Classifier (SVC), model `SVM\_Olson` won the battle at nearly 98.2% accuracy. This is not to say that this is the best model in all cases. All this means is that for the given test set, this model performed the best. Out of the 20% of data withheld in this test (114 random individuals), only a handful were misdiagnosed from all models. No model is perfect, but I am happy to see how well the recommendations from Olson worked on this data set. If on average less than a handful of people out of 114 are misdiagnosed with such accuracy and precision, that is a good start for making a model.

Receive Operating