Machine Learning Course Final Project:

Breast Cancer Diagnosis (link)

By: Tohar Tsvitman

ID: 318192838

Introduction

Breast cancer is the most commonly occurring cancer in women (even though men can have it too) and the second most common cancer overall.

It's been proven that early diagnosis and treatment increase the recovery rate.

In this project I attempt to classify breast lumps to malignant tumors which require a fast treatment and can be life risking, and benign lumps which can increase the risk of developing breast cancer later but are not immediately life risking.

The Data

The data I chose to this project contains information about breast tumors.

Each tumor has 30 different features and a diagnosis of whether it is malignant or benign

malignant count: 357 (62.7 %) benign count: 212 (37.3 %)

The features:

- ID number: the id number of the person with the tumor.
- Diagnosis: The diagnosis of breast tissues (M = malignant, B = benign).
- Radius mean: mean of distances from the center to points on the perimeter.
- Texture mean: standard deviation of gray-scale values (from the imaging).
- Perimeter mean: mean size of the core tumor.
- Area mean: mean area of the tumor.
- Smoothness mean: mean of local variation in radius lengths.
- Compactness mean: mean of perimeter^2 / area - 1.0.
- Concavity mean: mean of severity of concave portions of the contour.
- Concave points mean: mean for number of concave portions of the contour.
- Symmetry mean: mean of the tumor's symmetry.
- Fractal dimension mean: mean for "coastline approximation" – 1.

- Radius S.E: standard error for the mean of distances from center to points on the perimeter.
- Texture S.E: standard error for standard deviation of gray-scale values.
- Perimeter S.E: standard error for standard deviation of size of the core tumor.
- Area S.E: standard error for standard deviation of area of the tumor.
- Smoothness S.E: standard error for local variation in radius lengths.
- Compactness S.E: standard error for perimeter^2 / area - 1.0
- Concavity S.E: standard error for severity of concave portions of the contour
- Concave points S.E: standard error for number of concave portions of the contour
- Symmetry S.E: standard error of symmetric.
- Fractal dimension S.E: standard error for "coastline approximation" – 1.
- Radius worst: "worst" or largest mean value for mean of distances from the center to points on the perimeter.
- Texture worst: "worst" or largest mean value for standard deviation of gray-scale values.
- Perimeter worst: "worst" or largest mean value for standard deviation of size of the core tumor.
- Area worst: "worst" or largest mean value for standard deviation of the tumor area.
- Smoothness worst: "worst" or largest mean value for local variation in radius lengths.
- Compactness worst: "worst" or largest mean value for perimeter^2 / area - 1.0.
- Concavity worst: "worst" or largest mean value for severity of concave portions of the contour.
- Concave points worst: "worst" or largest mean value for number of concave portions of the contour.
- Symmetry worst: "worst" or largest mean value for number of tumor symmetric.
- Fractal dimension worst: "worst" or largest mean value for "coastline approximation" – 1

All those features are based on the tumor's imaging.

I changed the diagnosis to 0 for benign and 1 for malignant (instead of 'M' and 'B').

The patient ID is irrelevant, so I removed it from the data.

Different features have different ranges, for example, 'area_worst' is in range (100, 5000) and 'symmetry_se' is in range (0.007, 0.07). Therefore, I normalized the data using 'standard scalar'.

Some explanations

When giving someone a negative diagnosis i.e., that their lump is benign while in fact it is malignant is referred to as a **false negative**. The opposite case, i.e., diagnosing a lump as malignant while it is benign, is referred to as a **false positive**. When dealing with breast cancer, while giving someone a false positive diagnosis may be frightening, it will be found out quickly with further testing that the lump was in face benign.

On the other hand, giving someone a false negative diagnosis is dangerous and may be life threatening, since the disease will continue to develop without treatment. For this reason, I chose to score the different diagnosis algorithms with the F_β score, with $\beta = 0.5$, hence giving more weight to the recall measure.

The second thing I did was changing the class weight from "balanced" to a weight I found that gave the best F_β score, of course I gave class '1' (malignant) a higher weight.

To avoid overfitting, I used L2 regulation (ridge) and found optimal c rate using trial and error method.

Running the machine learning algorithms

1. Gaussian Naive Bayes

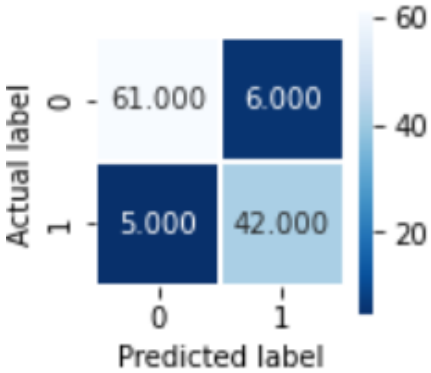
(No special adjustment)

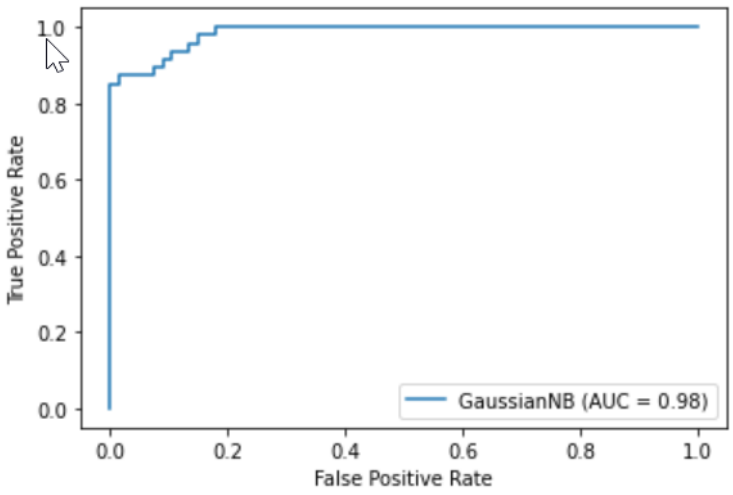
Number of mislabeled points out of a total 114 points: 11

Accuracy Score: 0.9035087719298246

FPR: 0.125

TPR: 0.9242424242424242





AUC: 0.9020323912353128

2. Logistic Regression:

Adjustment: find optimal c (0.1), to avoid overfitting, I reduce the range of θ using l2 regulation with the c I found

Find optimal classes weight {0 : 0.4 , 1 : 0.61}. gave 0 class less weight to reduce FN

(Trial and error method, best F_β score)

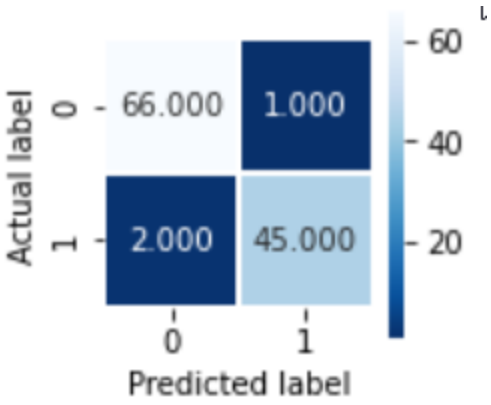
Number of mislabeled points out of a total 114 points: 3

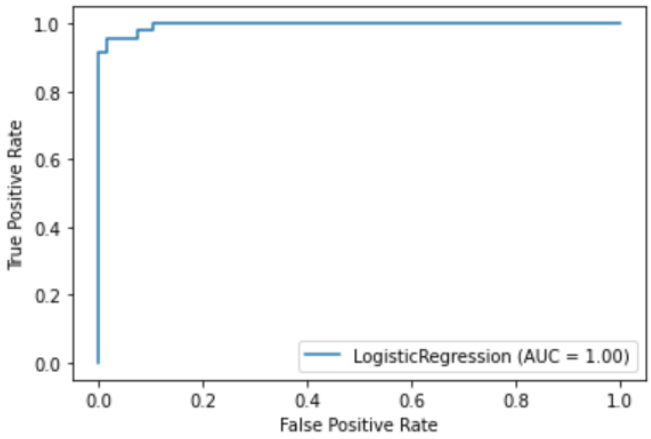
Accuracy Score: 0.9736882431339813

FPR: 0.014925373134328358

TPR: 0.9574468085106383

AUC: 0.971260717688155





3. AdaBoost Classifier

Adjustment: find optimal learning rate (1) (trial and error method, best F_β score) learning rate is the Weight applied to each regressor at each boosting iteration. A higher learning rate increases the contribution of each regressor

find optimal n_estimators (40) (trial and error method, best F_β score). The maximum number of estimators at which boosting is terminated. Estimator can be use more than once.

To avoid overfitting, I left the default base estimator in which max_depth = 3

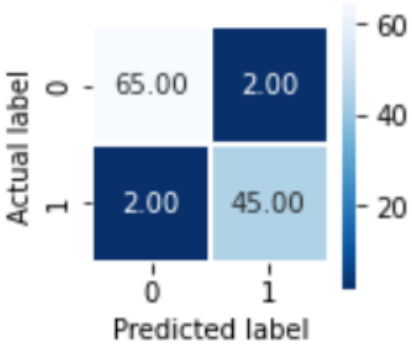
Number of mislabeled points out of a total 114 points: 4

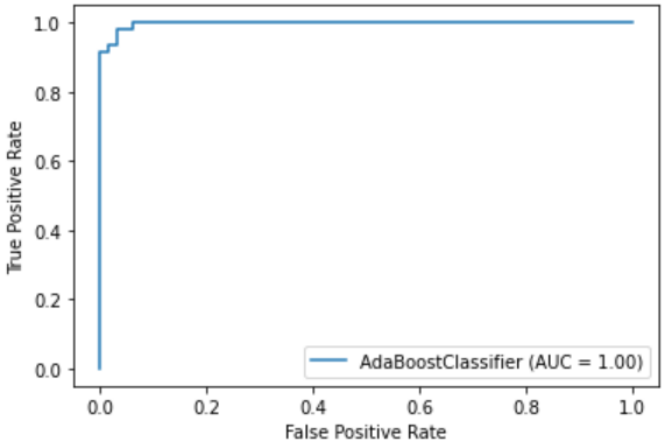
Accuracy Score: 0.9649122807017544

FPR: 0.029850746268656716

TPR: 0.9574468085106383

AUC: 0.9637980311209908





4. Linear Support Vector Machine

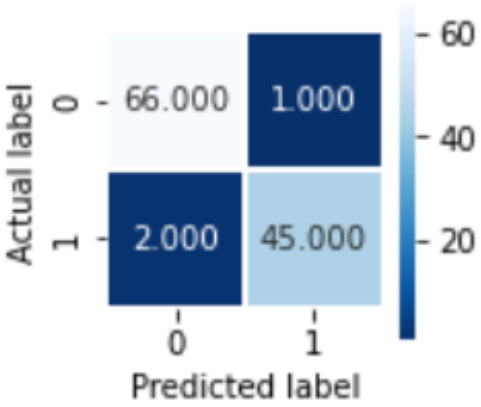
Number of mislabeled points out of a total 114 points: 3

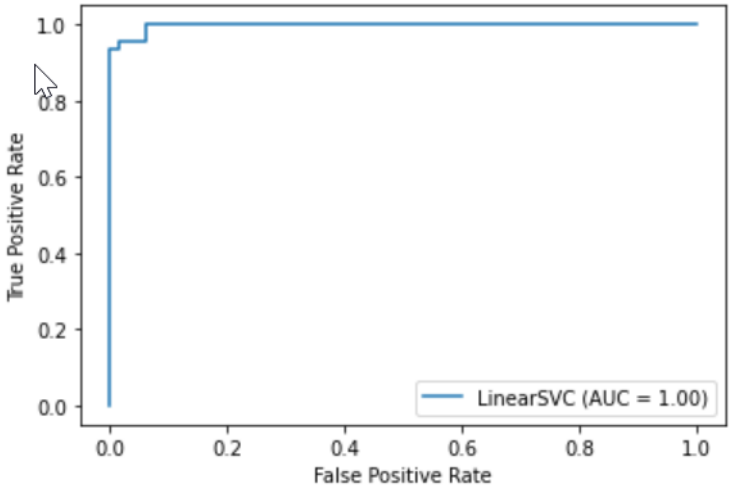
Accuracy Score: 0.9736882431339813

FPR: 0.021739130434782608

TPR: 0.9705882352941176

AUC: 0.971260717688155



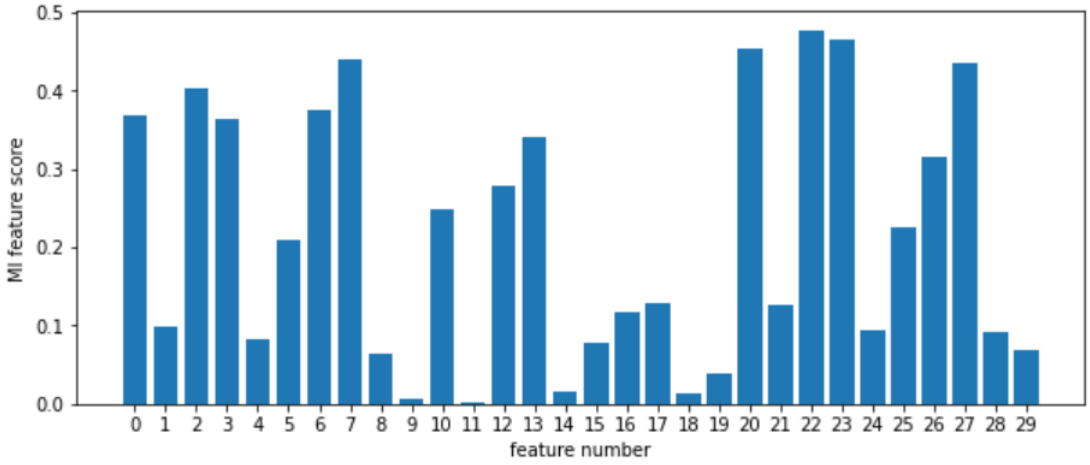


Feature selection:

I used 2 different ways to select most important features: Mutual information algorithm, and feature selection for a specific model.

1. Mutual information (MI):

The features rate according to MI algorithm:



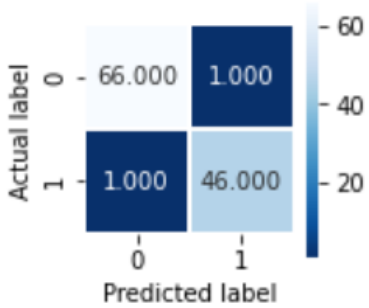
importance order: [22 23 20 7 27 2 6 0 3 13 26 12 10 25 5 17 21 16 24 1 28 4 15 29 8 19 14 18 9 11]

next I wanted to run logistic regression on reduced data. To do so I found how many features is best to select from the data when using logistic regression. I found out that 20 features will give the best results, so I took the 20 most important features from the data and ran the algorithm on the new data.

logistic regression result:

Number of mislabeled points out of a total 114 points: 2

Accuracy Score: 0.9824561403508771



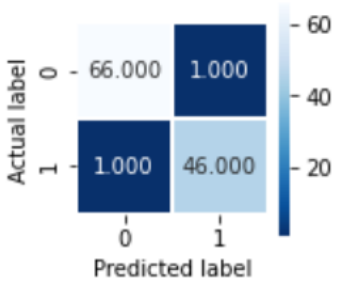
2. Feture selection from model:

2.1 logistic regression:

When using Logistic Regression algorithm the most important features are: [0, 1, 2, 3, 6, 7, 10, 12, 13, 20, 21, 22, 23, 24, 26, 27, 28]

Number of mislabeled points out of a total 114 points: 4

Accuracy Score: 0.9824561403508771



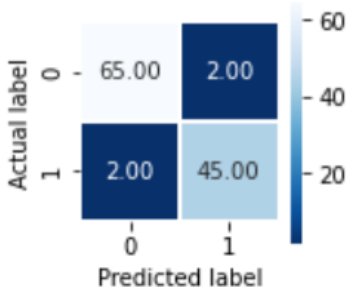
(verry similar result to the feature selection by MI algorithm. The only difference is here we have lass features)

2.2 AdaBoost

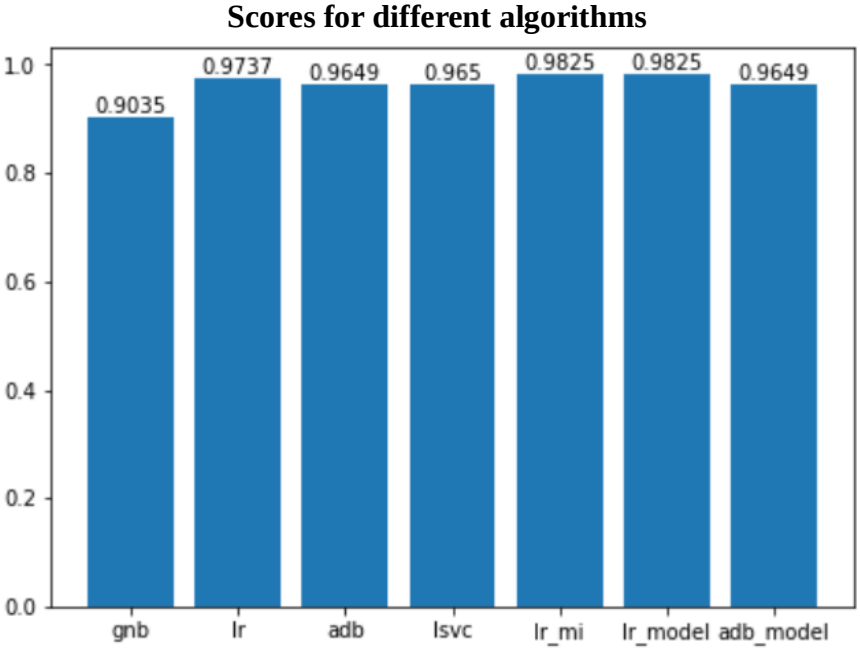
When using AdaBoost algorithm the most important features are: [1, 4, 5, 7, 11, 12, 13, 15, 18, 19, 21, 22, 23, 26, 27]

Number of mislabeled points out of a total 114 points: 4

Accuracy Score: 0.9649122807017544



Summary



The best result was achieved by using feature selection and the logistic regression algorithm. As can be seen above, the feature selection methods (model, mutual information) did not affect the results while using logistic regression; Although the selected features were different the score was the same.

It is possible that using a bigger or different dataset would yield different results when using different feature selection methods (in my estimation, choosing feature selection from model would yield better results than mutual information).

Overall, the results were good, and with little variation. The algorithm with highest score was also the one with the lowest false negative rate. Since having this quality is crucial when working with this kind of data, I believe this algorithm has a big advantage over the rest, even though they have a similar score.

The algorithm with the lowest score is the gaussian naïve base (gnb). I believe it might be because I did not make any adjustments to it, and if I wouldn't have don’t adjustments to the rest of the algorithms, their scores would also be lower.