Machine Learning Engineer Nanodegree

Capstone Project

Jeyvison Nascimento

May 28th, 2017

I. Definition

Project Overview

Breast cancer is the most dangerous type of cancer,besides lung cancer,a woman may develop. It's death rate is high and it is expected that 12% of the american women will develop it during their lifetime. The dataset to this problem was obtained from Wisconsin Breast Cancer Database.

Problem Statement

The goal is to create a program to help the doctor choose a protocol to deal with breast cancer when it's malignant or benign.Many informations will be considered like radius, perimeter and smoothness. All these informations come from a digitized image of a fine needle aspirate (FNA) of a breast mass image.With those information the cancer will be classified as malignant or benign and the doctor can decide which steps to take next

The program will:

- Train the model based on a dataset
- Identify if the cancer is malignant or benign

Metrics

At first, accuracy was considered as a metric to this problem but it gets distorted when the dataset is very unbalanced, as it is in this case.The metric used in the project is F-score. F-score is a harmonic mean between two other metrics called precision and recall. It allows to get more reliable metric results than accuracy because it handles well unbalanced datasets.

Precision can be described by the following formula:

$$Precision = \frac{\Sigma\, True\ positives}{\Sigma\, True\ Positives + \Sigma\, True\ Negatives}$$

Recall can be defined by the following formula:

$$Recall = \frac{\Sigma\, True\ Positives}{\Sigma\, True\ Positives + \Sigma\ False\ Negatives}$$

F-Score can be defined by the following formula:

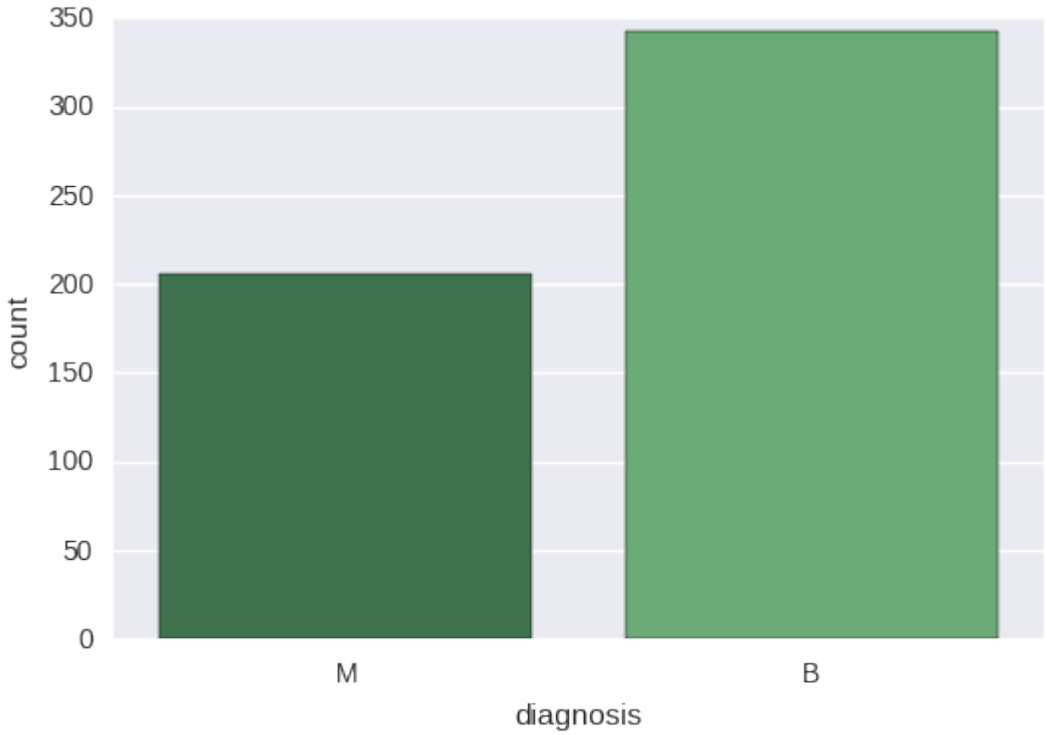
$$F - Score = \frac{2 * (Precison * Recall)}{(Precision + Recal)l}$$

II. Analysis

Data Exploration

The dataset that will be used has 30 features and a binary target feature that tells if the cancer was malignant(M) or benign(B). The input of the program will be a csv containing values for classification.

The dataset has 549 instances and is unbalanced ,as we can see in the graph below, so it will need to be stratified.



Also, all the features are real-valued with different ranges what makes us need to scale the dataset before training and during evaluation.

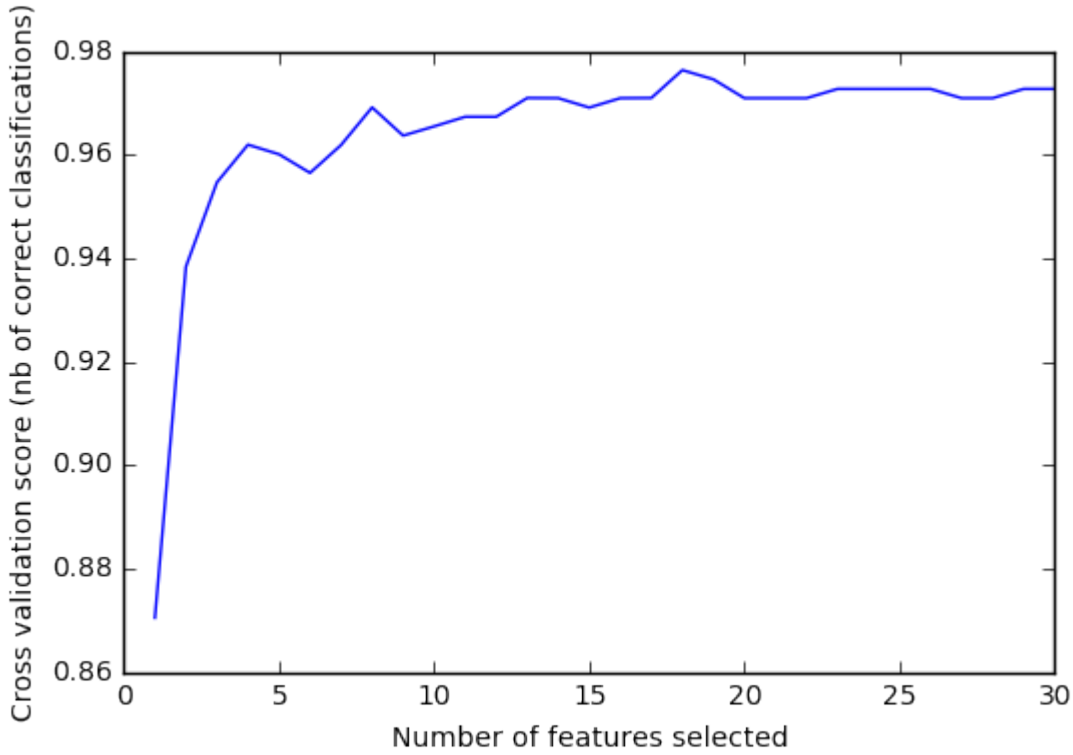
Here follows the list of the 30 features:

1. radius_mean
2. texture_mean
3. perimeter_mean
4. area_mean
5. smoothness_mean
6. compactness_mean
7. concavity_mean
8. concave points_mean
9. symmetry_mean
10. fractal_dimension_mean
11. radius_se
12. texture_se
13. perimeter_se
14. area_se
15. smoothness_se
16. compactness_se
17. concavity_se
18. concave points_se
19. symmetry_se
20. fractal_dimension_se
21. radius_worst
22. texture_worst

23. perimeter_worst
24. area_worst
25. smoothness_worst
26. compactness_worst
27. concavity_worst
28. concave points_worst
29. symmetry_worst
30. fractal_dimension_worst

Exploratory Visualization

Running RFECV we can find the optimal numbers of features as we can check in the graph below:



Algorithms and Techniques

The algorithm chosen for this problem is SVC. SVC is a SVM classification algorithm that performs very well when the number of data points is not very high which is the case here.It works with 'kernels' which are different decision functions and each kernel can be tuned by a specific parameter like **C, degree,gamma** or **coef0**. SVC,

when used with fold validation, has a decent probabilistic calculation which can be used to verify the likelihood of an output.

Also, RFECV(Recursive Feature Elimination with Cross Validation) is a feature selection algorithm that will help to reduce the number of features in our dataset, selecting only the most relevants. It select the features considering their initial weight set by the classifier, prune the ones with smallest weights , train and score in this subset and keeps going ‘till it reaches the best score it can(considering the number of features).

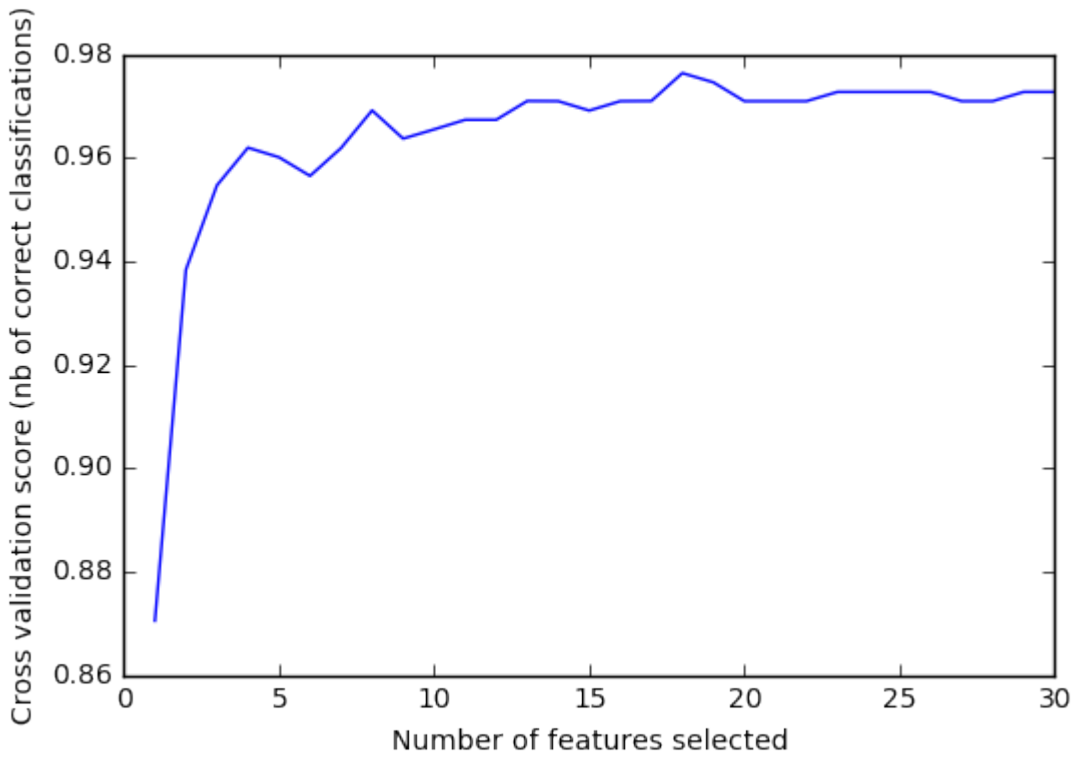
Benchmark

For benchmarking purposes we will use scikit learning Dummy classifier with stratified strategy(the default one). After the training, the F-Score metric will be used to measure the predicted output from each classifier.

III. Methodology

Data Preprocessing

As said before, the data needs to be scaled and have its dimensionality reduced so we may retrieve reliable outputs from it. The data was first scaled using log function. Some values turned “-Inf” so they were set to 0. Then, RFECV was used to reduce the number of features we needed as you can see in the graph below:



F-Score score was used as a metric to find the best number of features.

Implementation

First, the SVC classifier and the dummy classifier will be trained with the preprocessed data using StratifiedKFold to assure balance. The Evaluation data will be loaded,scaled and reduced, as the original data was, and then the predictions and the f-score score will be output.

After the split the training test set had 492 points and the test dataset has 57 points.

There was no complication on splitting the data.

Refinement

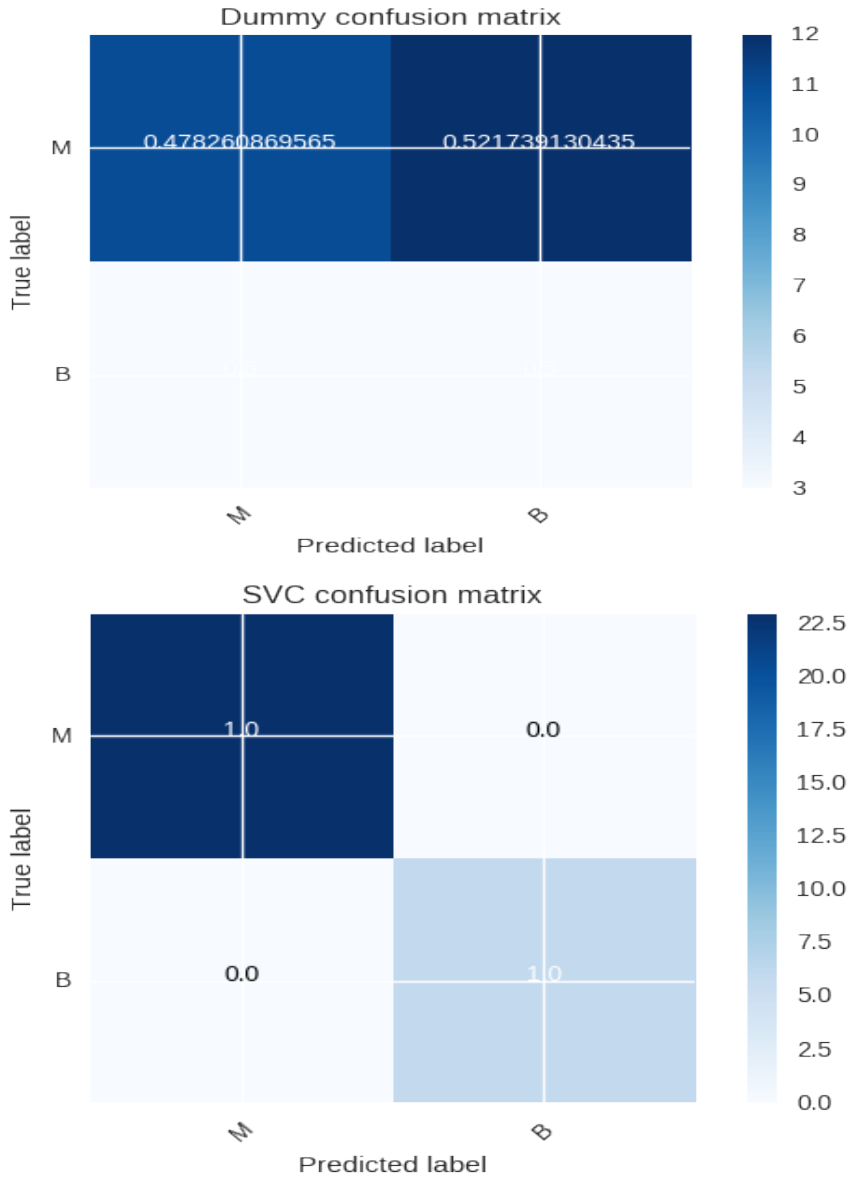
No requirements were needed to successfully train and test the classifier.Trainings with all the features showed the same result score.

.

IV. Results

Model Evaluation and Validation

As we can check by the F-Score result the model is generalizing very well and its output can be trustful. We can check the below graphs that shows the confusion matrix for SVC and Dummy classifier.



Justification

The F-Score Dummy classifier was 0.53 and the F-score result for our SVC classifier was 1.0 so we prove by that that our model is well trained and is better than dummy classifying.

V. Conclusion

Free-Form Visualization

An important quality about the project is the high F-Score considering data never seen.

Reflection

The scaling part was a little tough. First i tried to use Box-Cox but there are some non positives values in the dataset i’m using. Then i tried to use log and some values turned to ‘-Inf’ which can’t be used when so for all ‘-Inf’ values i setted 0.

Improvement

An improvement that can be made is use PCA to reduce even more the dimensional feature space to some value that still represents the variance we have in the data. Parameter tuning wasn’t used too, which can be useful if the dataset get’s bigger