Breast cancer prediction

Nikola Maksimovic SW15/2016

1. Motivation

Breast cancer (BC) is one of the most common cancers among women worldwide, representing the majority of new cancer cases and cancer-related deaths according to global statistics, making it a significant public health problem in today's society. The early diagnosis of BC can improve the prognosis and chance of survival significantly, as it can promote timely clinical treatment to patients. Further accurate classification of benign tumors can prevent patients undergoing unnecessary treatments. Thus, the correct diagnosis of BC and classification of patients into malignant or benign groups is the subject of much research. Because of its unique advantages in critical features detection from complex BC datasets, machine learning (ML) is widely recognized as the methodology of choice in BC pattern classification and forecast modelling. Classification and data mining methods are an effective way to classify data. Especially in medical field, where those methods are widely used in diagnosis and analysis to make decisions.

2. Research questions

There were only two questions:

1. Is there any null/fault values in the data set?
2. Which model fits the best?

I wanted to make the predictions be as precise as they can, because good prediction can save someone's life if it's predicted in a early phase. Data set I was working with is the data set from [1], which is live data from some of the biggest clinics in world. The data set consists of 33 columns that helps prediction work better:

- Diagnosis (M = malignant, B = benign)
- Radius_mean : Mean of distances from center to points on the perimeter
- Texture_mean : standard deviation of gray-scale values
- Perimeter_mean : mean size of the core 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
- Fractal_dimension_mean : mean for "coastline approximation" - 1
- Radius_se : standard error for standard deviation of gray-scale values
- Smoothness_se : standard error for local variation in radius lengths
- Compactness_se : standard error for perimeter^2 / area - 1.0
- Concavity_se : standard error for severity of concave portions of the contour
- Fractal_dimension_worst : "worst" or largest mean value for "coastline approximation" –
- …

3. Related work

There is a lot of code on the Internet that is solving the exact same problem as me with same data set I used, with many different approaches, with different types of classifiers, and with different validations. The problem is big, and the solutions are different, but I realized they aren't so precise, that's why I tried to improve it even more.

4. Methodology

My solution consisted of couple steps:

1. **Data exploration** – see what exact columns data set has, it's shape, which one should I use for X, and which one for Y variable.

2. **Data preparation** – importing necessary libraries, finding null values and deleting the column with them, label encoding (diagnosis column has String type)
3. **Plotting** – did plotting in order to see correlation between columns (libraries seaborn and matplotlib), and also to see the number of Benign vs Malignant cases in data set
4. **Splitting the data set** – into train and test data sets (80:20)
5. **Scalling** – using StandardScaler to scale values between 0-1
6. **Model making and choosing the best model** – did parallel comparison of LogisticRegression, DecisionTreeClassifier and RandomForestClassifier and check which one is the best using KFold cross validation

5. **Discussion**

Model performances did vary a litle bit, but all of the models were doing the predictions well on given data set. I was using KFold cross validation and the results are:

- **Logistical regression** has accuracy score of 0.95 which is very good score.
- **DecisionTreeClassifier** has smaller score than the LogisticalRegression which is 0.94
- **RandomForestClassifier** has the biggest score of all three and it is equal to 0.96

Comparing the results of every model in multiple tests, I chose the RandomForestClassifier the best of these 3.

6. **References**

[1] **Data set** - https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/

[2] **Scikit-learn** - https://scikit-learn.org/stable/

[3] **Kaggle** - https://www.kaggle.com/