Breast Cancer Prediction Models Research

Document Content

Author Armen Melkumyan

Introduction

Breast cancer is a prevalent cause of death, and it is the only type of cancer that is widespread among women worldwid. Many imaging techniques have been developed for early detection and treatment of breast cancer and to reduce the number of deaths, and many aided breast cancer diagnosis methods have been used to increase the diagnostic accuracy.

In the last few decades, several data mining and machine learning techniques have been developed for breast cancer detection and classification, which can be divided into three main stages: preprocessing, feature extraction, and classification. To facilitate interpretation and analysis, the preprocessing of mammography films helps improve the visibility of peripheral areas and intensity distribution, and several methods have been reported to assist in this process.

Feature extraction is an important step in breast cancer detection because it helps discriminate between benign and malignant tumors. After extraction, image properties such as smoothness, coarseness, depth, and regularity are extracted by segmentation

Dataset Used for Research

In this work, Breast Cancer dataset was obtained from the UCI Machine Learning Repository.

Preprocessing¶

As a part of this research, processing was performed on the raw breast cancer data to scale the features using the Standard Scaler module. Standardization of datasets is a common requirement for many machine learning estimators. It transforms the attributes to a standard Gaussian distributions based on (xi–mean(x))/stdev(x) where stdev is the standard deviation. The Robust Scaler depends on the interquartile range to transform the features using (xi–Q1(x))/(Q3(x)–Q1(x)), where Q1, Q2, and Q3 represent quartiles. All the transformations used are included in scikit-learn machine learning library

Features Selection

Usually, feature selection is applied as a preprocessing step before the actual learning. However, no algorithm can make good predictions without informative and discriminative features; therefore, to keep the most significant features and reduce the size of the dataset, we implemented Kernel PCA.

Machine Learning Algorithms

For this research I have used KNN, SVM, KSVM, Decission Tree, Random Forest, Logistic Regression, XGBoost algorithms.

Dataset Structure

	Age	BMI	Glucose	Insulin	HOMA	Leptin	Adiponectin	Resistin	MCP.1
0	48	23.500000	70	2.707	0.467409	8.8071	9.702400	7.99585	417.114
1	83	20.690495	92	3.115	0.706897	8.8438	5.429285	4.06405	468.786
2	82	23.124670	91	4.498	1.009651	17.9393	22.432040	9.27715	554.697
3	68	21.367521	77	3.226	0.612725	9.8827	7.169560	12.76600	928.220
4	86	21.111111	92	3.549	0.805386	6.6994	4.819240	10.57635	773.920
5	49	22.854458	92	3.226	0.732087	6.8317	13.679750	10.31760	530.410
6	89	22.700000	77	4.690	0.890787	6.9640	5.589865	12.93610	1256.083
7	76	23.800000	118	6.470	1.883201	4.3110	13.251320	5.10420	280.694
8	73	22.000000	97	3.350	0.801543	4.4700	10.358725	6.28445	136.855
9	75	23.000000	83	4.952	1.013839	17.1270	11.578990	7.09130	318.302
10	34	21.470000	78	3.469	0.667436	14.5700	13.110000	6.92000	354.600
11	29	23.010000	82	5.663	1.145436	35.5900	26.720000	4.58000	174.800
12	25	22.860000	82	4.090	0.827271	20.4500	23.670000	5.14000	313.730
13	24	18.670000	88	6.107	1.330000	8.8800	36.060000	6.85000	632.220

Data Splitting

```
from sklearn.model_selection import train_test_split
```

```
X_trainset, X_testset, y_trainset, y_testset = train_test_split(X, y, test_size=0.20, random_state=0)
```

```
print(X_trainset.shape)
print(y_trainset.shape)
```

```
(92, 9)
(92, 1)
```

```
print(X_testset.shape)
print(y_testset.shape)
```

```
(24, 9)
(24, 1)
```

Dimenson Reduction

Kernel PCA

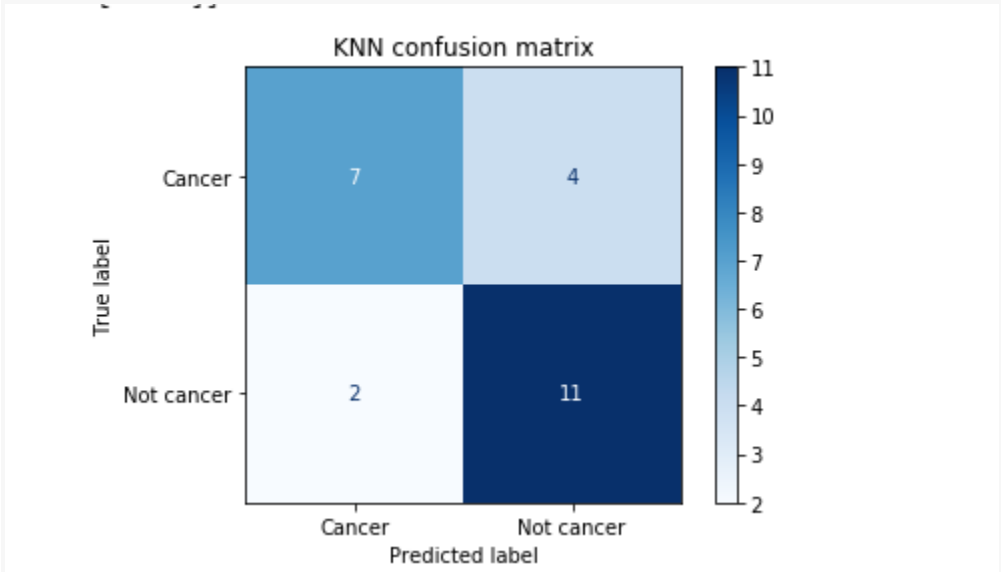
Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.

```
# Applying Kernel PCA
from sklearn.decomposition import KernelPCA
kpca = KernelPCA(n_components = 7, kernel = 'rbf')
X_trainset = kpca.fit_transform(X_trainset)
X_testset = kpca.transform(X_testset)
```

KNN Result

```
precision_score : 0.6363636363636364
recall_score: 0.7777777777777778
f1_score: 0.7000000000000001
```

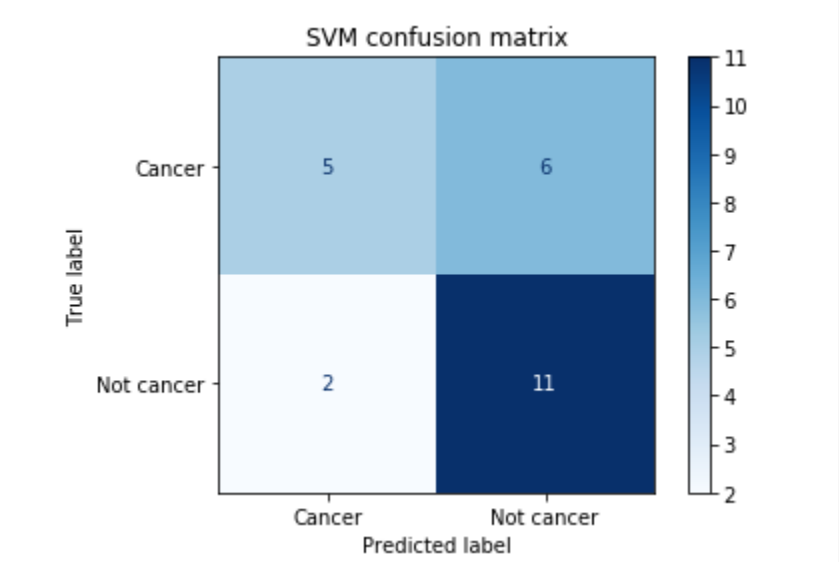
```
KNN Confusion Matrix
```



SVM Result

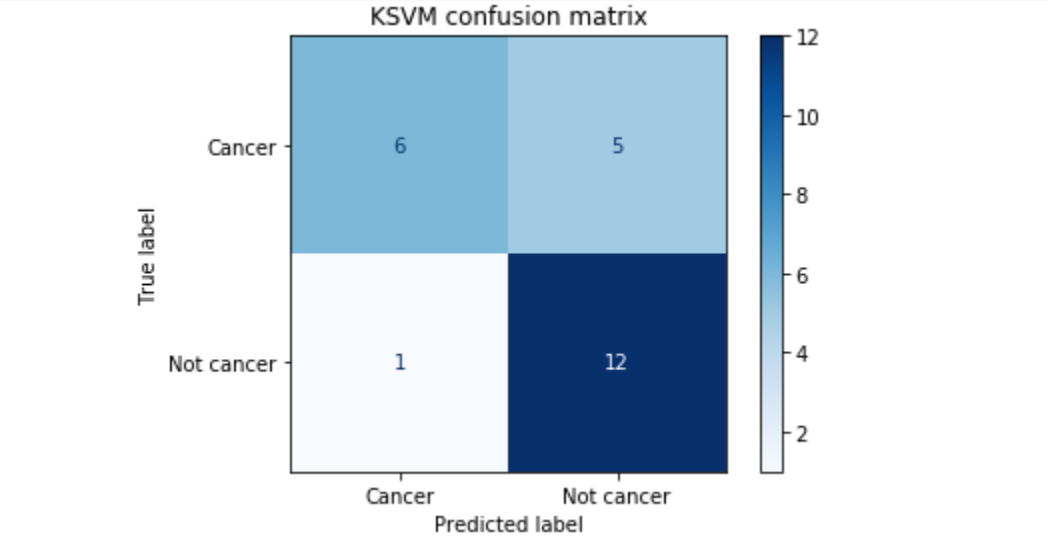
```
precision_score : 0.45454545454545453
recall_score: 0.7142857142857143
f1_score: 0.5555555555555556
```

```
SVM Confusion Matrix
```



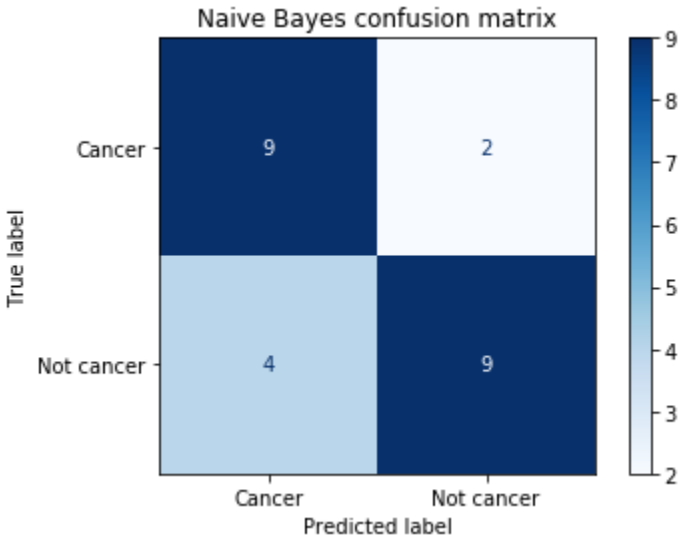
Kernel SVM Result

```
precision_score : 0.5454545454545454
recall_score: 0.8571428571428571
f1_score: 0.6666666666666665
Kernel SVM Confusion Matrix
```



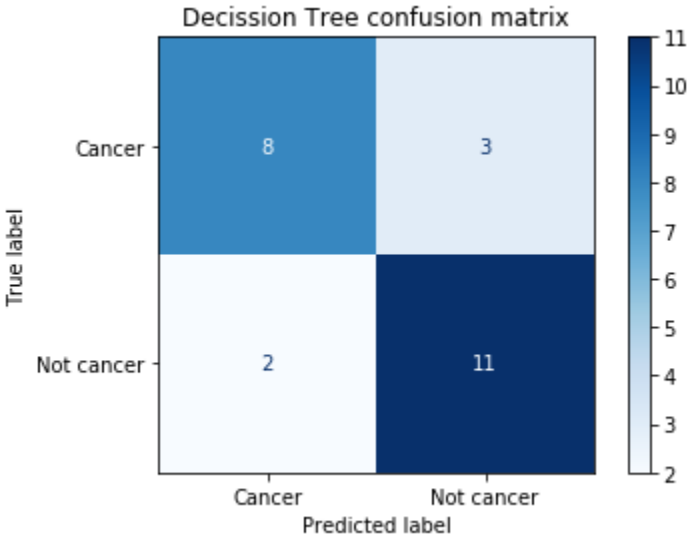
Naive Bayes Result

```
precision_score : 0.8181818181818182
recall_score: 0.6923076923076923
f1_score: 0.7500000000000001
Naïve Bayes Confusion Matrix
```



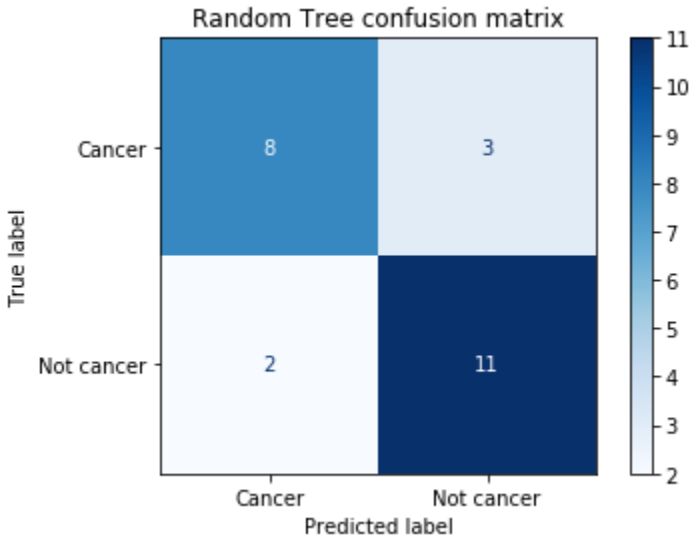
Decission Tree Result

```
precision_score : 0.7272727272727273
recall_score: 0.8
f1_score: 0.761904761904762
Decission Tree Confusion Matrix
```



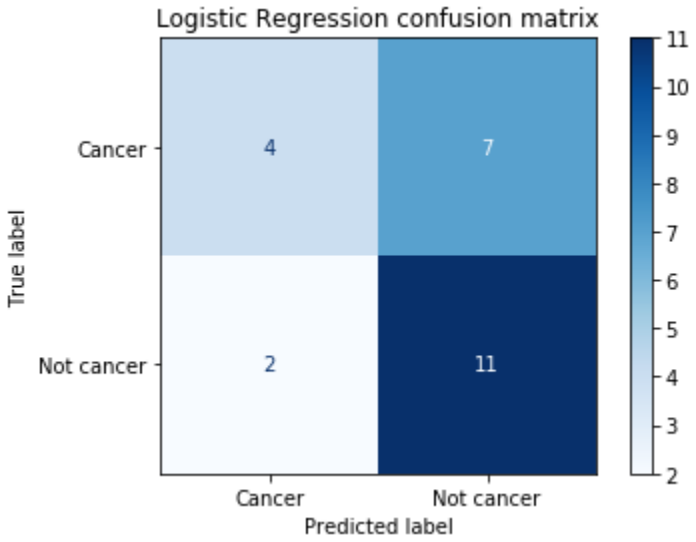
Random Forest Result

```
precision_score : 0.7272727272727273
recall_score: 0.8
f1_score: 0.761904761904762
Random Forest Confusion Matrix
```



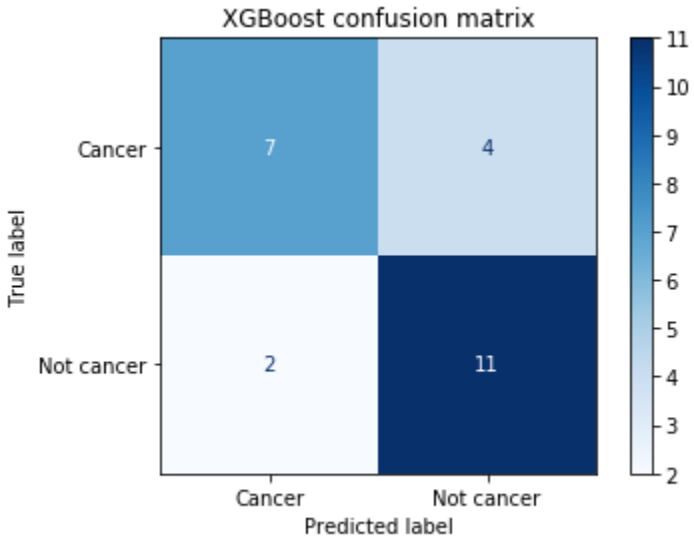
Logistic Regression Result

```
precision_score : 0.36363636363636365
recall_score: 0.6666666666666666
f1_score: 0.4705882352941177
Logistic Regression Confusion Matrix
```



XGBoost Result

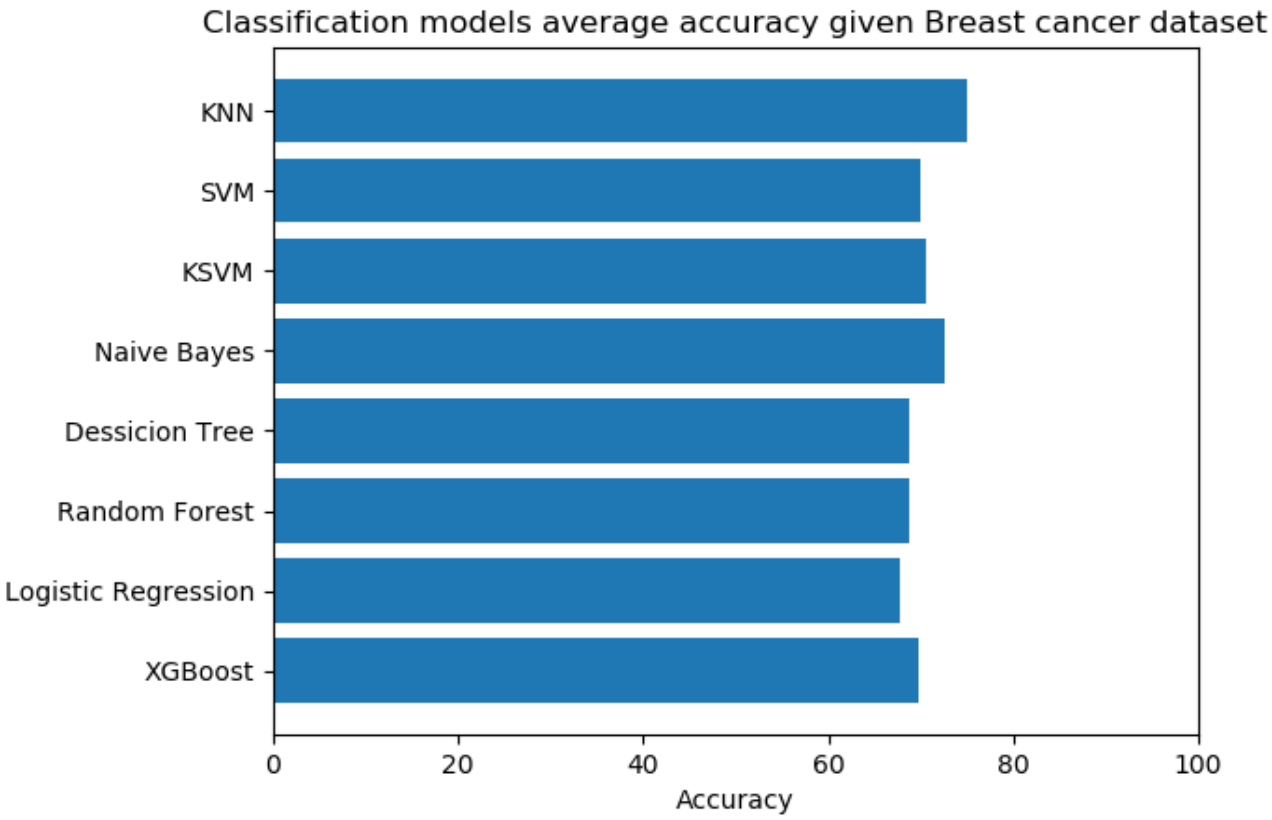
precision_score : 0.6363636363636364
recall_score: 0.7777777777777778
f1_score: 0.7000000000000001
XGBoost Confusion Matrix



Output all algorithms acuracy means

```
knn_acuracy: 0.75
svm_acuracy: 0.6988888888888889
ksvm_acuracy: 0.7066666666666667
naive_bayes_acuracy: 0.7266666666666667
tree_acuracy: 0.6866666666666668
random_tree_acuracy: 0.6866666666666666
logistic_acuracy: 0.6777777777777778
xgboost_cm: 0.6966666666666667
```

Classification models average accuracy given Breast cancer dataset



Conclussion

As we can see KNN is the best choose for predict Breast cancer with given dataset. If we will have bigger dataset of course our prediction precision will be much higher.

Author Armen Melkumyan