

# **Final Project Report**

## **Breast Cancer Classification**

*Saurabh Shrinivas Maydeo*

*27<sup>th</sup> April 2020*

# Business Understanding

## Business Problem

Breast cancer is the top cancer in women both in the developed and the developing world. The incidence of breast cancer is increasing in the developing world due to increased life expectancy, increased urbanization and adoption of western lifestyles. Although some risk reduction might be achieved with prevention, these strategies cannot eliminate the majority of breast cancers that develop in low- and middle-income countries where breast cancer is diagnosed in very late stages. Therefore, early detection in order to improve breast cancer outcome and survival remains the cornerstone of breast cancer control. It is practically impossible for doctors to look at the data of each patient for breast cancer prediction. Thus, the business problem here is, to automate the task of classify the patients into 2 groups – malignant and benign for breast cancer, so that the doctors would only have to cross check the data about those patients who are classified as malignant for breast cancer.

## Dataset

For this problem, we will be using Breast Cancer Wisconsin (Diagnostic) Data Set provided by UCI Machine Learning Repository. Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image.

This data set is created by Dr. William H. Wolberg from University of Wisconsin, W. Nick Street from University of Wisconsin and Olvi L. Mangasarian from University of Wisconsin.

Attribute Information:

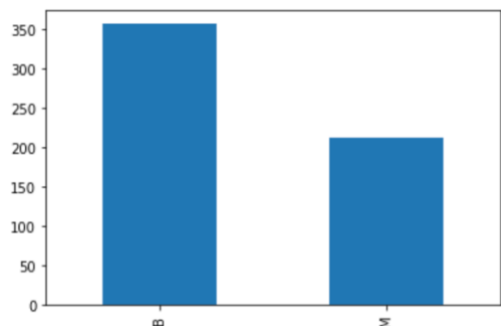
- Link to the data set:  
<https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29>
- Data Set Characteristics: Multivariate
- Number of attributes: 32
- Number of missing values: 0
- Class distribution: 357 benign, 212 malignant
- Target Variable: The diagnosis of breast tissues (M = malignant, B = benign)

## Proposed Analytics Solution

We can train machine model on the previous data about the patients who had and didn't have breast cancer, so that it learns to classify the new patient based upon his or her data. This will help in early diagnosis of breast cancer which can improve the chances of patients getting good treatment from the beginning and thus increasing chances of their survival.

# Data Exploration and Preprocessing

## Distribution of the target variable



### Data Quality Report

	feature	count	missing %	unique values	mean	std	min	Q1	median	Q3	max	IQR
0	radius_mean	569	0.000000	456	14.127292	3.524049	6.981000	11.700000	13.370000	15.780000	28.110000	4.080000
1	texture_mean	569	0.000000	479	19.289649	4.301036	9.710000	16.170000	18.840000	21.800000	39.280000	5.630000
2	perimeter_mean	569	0.000000	522	91.969033	24.298981	43.790000	75.170000	86.240000	104.100000	188.500000	28.930000
3	area_mean	569	0.000000	539	654.889104	351.914129	143.500000	420.300000	551.100000	782.700000	2501.000000	362.400000
4	smoothness_mean	569	0.000000	474	0.096360	0.014064	0.052630	0.086370	0.095870	0.105300	0.163400	0.018930
5	compactness_mean	569	0.000000	537	0.104341	0.052813	0.019380	0.064920	0.092630	0.130400	0.345400	0.065480
6	concavity_mean	569	0.000000	537	0.088799	0.079720	0.000000	0.029560	0.061540	0.130700	0.426800	0.101140
7	concave points_mean	569	0.000000	542	0.048919	0.038803	0.000000	0.020310	0.033500	0.074000	0.201200	0.053690
8	symmetry_mean	569	0.000000	432	0.181162	0.027414	0.106000	0.161900	0.179200	0.195700	0.304000	0.033800
9	fractal_dimension_mean	569	0.000000	499	0.062798	0.007060	0.049960	0.057700	0.061540	0.066120	0.097440	0.008420
10	radius_se	569	0.000000	540	0.405172	0.277313	0.111500	0.232400	0.324200	0.478900	2.873000	0.246500
11	texture_se	569	0.000000	519	1.216853	0.551648	0.360200	0.833900	1.108000	1.474000	4.885000	0.640100
12	perimeter_se	569	0.000000	533	2.866059	2.021855	0.757000	1.606000	2.287000	3.357000	21.980000	1.751000
13	area_se	569	0.000000	528	40.337079	45.491006	6.802000	17.850000	24.530000	45.190000	542.200000	27.340000
14	smoothness_se	569	0.000000	547	0.007041	0.003003	0.001713	0.005169	0.006380	0.008146	0.031130	0.002977
15	compactness_se	569	0.000000	541	0.025478	0.017908	0.002252	0.013080	0.020450	0.032450	0.135400	0.019370
16	concavity_se	569	0.000000	533	0.031894	0.030186	0.000000	0.015090	0.025890	0.042050	0.396000	0.026960
17	concave points_se	569	0.000000	507	0.011796	0.006170	0.000000	0.007638	0.010930	0.014710	0.052790	0.007072
18	symmetry_se	569	0.000000	498	0.020542	0.008266	0.007882	0.015160	0.018730	0.023480	0.078950	0.008320
19	fractal_dimension_se	569	0.000000	545	0.003795	0.002646	0.000895	0.002248	0.003187	0.004558	0.029840	0.002310
20	radius_worst	569	0.000000	457	16.269190	4.833242	7.930000	13.010000	14.970000	18.790000	36.040000	5.780000
21	texture_worst	569	0.000000	511	25.677223	6.146258	12.020000	21.080000	25.410000	29.720000	49.540000	8.640000
22	perimeter_worst	569	0.000000	514	107.261213	33.602542	50.410000	84.110000	97.660000	125.400000	251.200000	41.290000
23	area_worst	569	0.000000	544	880.583128	569.356993	185.200000	515.300000	686.500000	1084.000000	4254.000000	568.700000
24	smoothness_worst	569	0.000000	411	0.132369	0.022832	0.071170	0.116600	0.131300	0.146000	0.222600	0.029400
25	compactness_worst	569	0.000000	529	0.254265	0.157336	0.027290	0.147200	0.211900	0.339100	1.058000	0.191900
26	concavity_worst	569	0.000000	539	0.272188	0.208624	0.000000	0.114500	0.226700	0.382900	1.252000	0.268400
27	concave points_worst	569	0.000000	492	0.114606	0.065732	0.000000	0.064930	0.099930	0.161400	0.291000	0.096470
28	symmetry_worst	569	0.000000	500	0.290076	0.061867	0.156500	0.250400	0.282200	0.317900	0.663800	0.067500
29	fractal_dimension_worst	569	0.000000	535	0.083946	0.018061	0.055040	0.071460	0.080040	0.092080	0.207500	0.020620

## Missing Values

There are no missing values in the data.

```
[17]: data.isnull().sum().sum()
```

```
Out[17]: 0
```

## Normalization

Some classifiers require normalized data to work on. We have performed minmax normalization for each feature of the data set except class variable. We have used MinMaxScaler from the sklearn's preprocessing package.

*Note: We will be feeding normalized data to all classifiers except Random Forest Classifier.*

## Feature Selection and Transformations

To avoid the overfitting caused by the curse of dimensionality, we have performed feature selection. In the original data set, there are total 32 features. After removing the Unnamed:32 feature which doesn't have any values, there are still 31 features. We have used chi2 test to select k=10 best features.

These are the 10 best features selected after ch2 test. There is off-course class variable 'diagnosis' with us.

```
Out[52]: array(['radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean',  
        'perimeter_se', 'area_se', 'radius_worst', 'texture_worst',  
        'perimeter_worst', 'area_worst'], dtype=object)
```

We have mapped our class variable – diagnosis (B, M) into 0 and 1.

# Model Selection and Evaluation

## Evaluation Metrics

In this problem, precision is the most important metric. Other important metrics are f1 score, accuracy and recall. If 2 models are having the same accuracy, we would select the one with the higher precision as there shouldn't be any patient who has cancer and our model classifies him/her as benign for breast cancer.

## Models

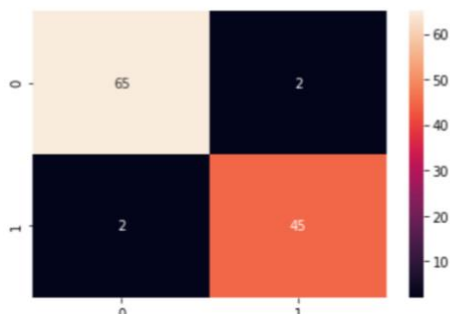
We have developed at least 1 model from each category of supervised learning algorithms.

### Information Based Learning: Random Forest Classifier

We have built Random Forest Classifier with criterion='gini', number of estimators=1000, max\_depth=3, random\_state=0, n\_jobs=-1.

```
Accuracy on test set is 0.9649122807017544
f1 score on test set is 0.9574468085106385
Precision on test set is 0.9574468085106383
Recall on test set is 0.9574468085106383
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa7d3d62748>
```



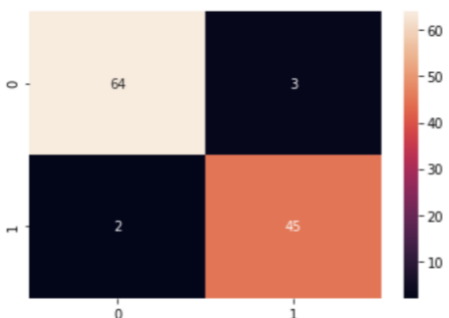
### Error Based Learning: SVM, Logistic Regression

#### 1. SVM

These are the results that SVM classifier gave on the test set.

```
Accuracy on test set is 0.956140350877193
f1 score on test set is 0.9473684210526315
Precision on test set is 0.9375
Recall on test set is 0.9574468085106383
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa7d45fc080>
```

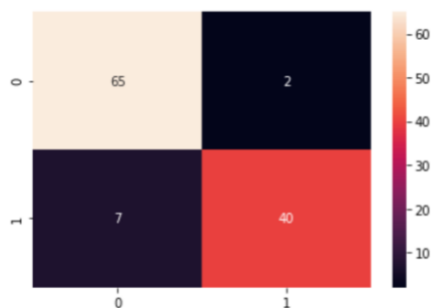


## 2. Logistic Regression

These are the results that Logistic Regression classifier gave on the test set.

```
Accuracy on test set is 0.9210526315789473  
f1 score on test set is 0.898876404494382  
Precision on test set is 0.9523809523809523  
Recall on test set is 0.851063829787234
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa7d38aeb38>
```

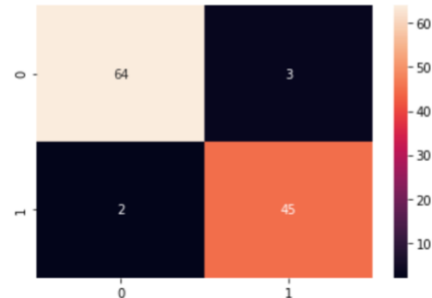


## Similarity Based Learning: kNN Classifier

We have built kNN classifier with k=19. These are the results that kNN classifier gave on the test set.

```
Accuracy on test set is 0.956140350877193  
f1 score on test set is 0.9473684210526315  
Precision on test set is 0.9375  
Recall on test set is 0.9574468085106383
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa7d34497b8>
```

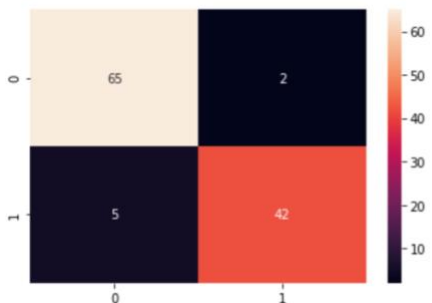


## Probability Based Learning: Naïve Bayes Classifier

We built Naïve Bayes classifier. These are the results that Naïve Bayes classifier gave on the test set.

```
Accuracy on test set is 0.9385964912280702  
f1 score on test set is 0.9230769230769231  
Precision on test set is 0.9545454545454546  
Recall on test set is 0.8936170212765957
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa7d32e9e10>
```



## Sampling and Evaluation Settings

We divided the entire data set into 3 parts – Training set, Validation set and Test set. Test is 20% of the entire data set, train set is 64% of the entire data set and validation set is 16% of the entire data set.

We performed cross-validation to make sure that our models aren't overfitting.

## Evaluation

Following is the table for comparing the evaluation metrics of all the models that we have built.

	Accuracy	f1 score	Precision	Recall
<b>RandomForest</b>	0.964912	0.957447	0.957447	0.957447
<b>SVM</b>	0.956140	0.947368	0.937500	0.957447
<b>LogisticRegression</b>	0.921053	0.898876	0.952381	0.851064
<b>kNN</b>	0.956140	0.947368	0.937500	0.957447
<b>Naive Bayes</b>	0.938596	0.923077	0.954545	0.893617

As discussed earlier, with all things being equal, the model that gives high precision is best choice for the breast cancer classification problem. Thus, Random Forest classifier is the best of model.

## Results and Conclusion

As discussed earlier, precision is the most important metric for evaluation in this task. Other very important metrics for evaluation are f1 score, accuracy and recall.

	Accuracy	f1 score	Precision	Recall
RandomForest	0.964912	0.957447	0.957447	0.957447

Random Forest classifier outperforms all other models in terms of precision, f1 score, accuracy and recall. Thus, we propose this model for this task.

The Early detection in order to improve breast cancer outcome and survival remains the cornerstone of breast cancer control. The proposed analytics model – Random Forest classifier seems to be promising model for this task after carefully comparing it with other models based on the selected metrics for evaluation.

Therefore, health care practitioners can make use of this model to classify patients into malignant and benign for breast cancer. After getting classification results from the model, they themselves can check those data to make sure it is correctly classified.