Breast Cancer Detection

Machine Learning

—

Yashashwi Shah

Dhatri Ramagiri

Gauri Kasar

Contribution

We have implemented three different models for breast cancer detection.Accuracies on the test data and confusion matrix is implemented. We divided these models one amongst each.Task implementation is as follows:

Logistic Regression : Gauri

KNN : Yashaswi

Naive Bayes Classifier : Dhatri.

Abstract:

Studies have shown 1 out of 8 women are diagnosed with breast cancer. Early detection of this disease has a good amount for survival.Machine learning models can be used to predict the result of the data. There are three main stages in this analysis to get accurate results namely data preprocessing, feature selection and classification. Of these feature selection is the most important step while building any machine learning model. Exposing our model to unnecessary and many features can affect its accuracy.

Our goal in this project was to implement different machine learning models on the breast_cancer_bd.csv dataset. We have calculated the accuracies and confusion matrix for each model on the test data.

Data Selection:

Dataset is an important parameter as the accuracy of the classifier depends on the input data. A good dataset should be chosen which has many features. In this project we have chosen Breast Cancer Wisconsin (Original) Data Set . There are 11 features in this dataset along with the classification label. The labels are classified as 2 and 4 in the dataset where 2 stands for Benign and 4 stands for Malignant. The dataset has a total 699 rows, out of which 458 rows are benign and 241 are malignant.

Below are the features used in the dataset.

- Clump Thickness
- Uniformity of Cell Size
- Uniformity of Cell Shape
- Marginal Adhesion
- Single Epithelial Cell Size

- Bare Nuclei
- Bland Chromatin
- Normal Nucleoli
- Mitoses
- Class

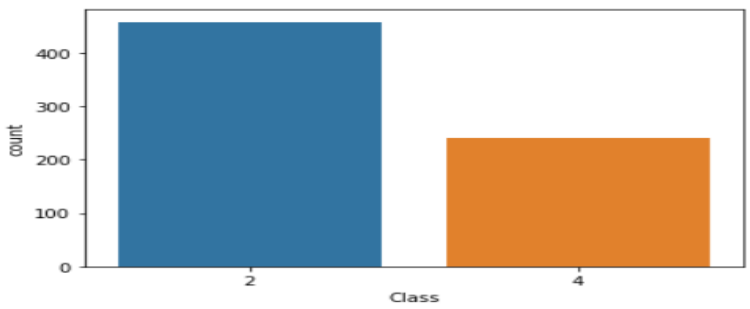
Sample code number	Clump Thickness	Uniformity of Cell Size	Uniformity of Cell Shape	Marginal Adhesion	Single Epithelial Cell Size	Bare Nuclei	Bland Chromatin	Normal Nucleoli	Mitoses	Class
1000025	5	1	1	1	2	1	3	1	1	2
1002945	5	4	4	5	7	10	3	2	1	2
1015425	3	1	1	1	2	2	3	1	1	2
1016277	6	8	8	1	3	4	3	7	1	2
1017023	4	1	1	3	2	1	3	1	1	2

Data Preprocessing:

Dataset is clean and it does not have any missing values. Data is split between target and predicted. Predicted are the ones that our model classifies as malign and benign and target are the values in the data set present that says whether the data is benign or malign. We have used standard library to show distribution of this values in the dataset.

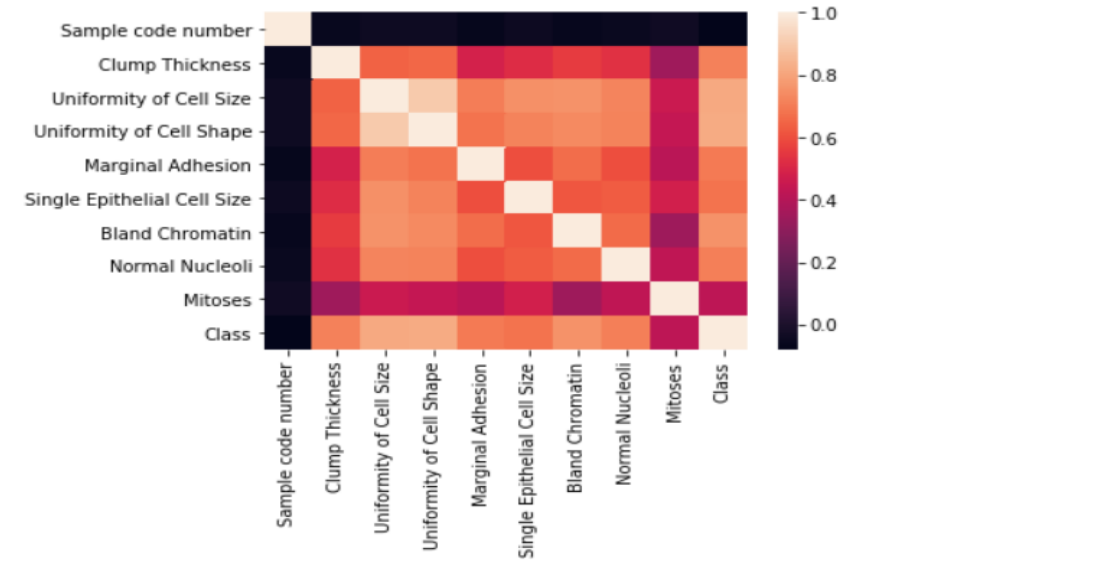
```
In [10]: import pandas as pd
         import seaborn as sns
         CSVFeedData = pd.read_csv("breast_cancer_bd.csv")
         y= CSVFeedData["Class"]
         sns.countplot(y)
         target_temp = CSVFeedData.Class.value_counts()
         print(target_temp)
```

```
2    458
4    241
Name: Class, dtype: int64
```



Correlation:

Correlation is when one value changes how it affects another value. We can use scatter plots or heat maps to view such relations between continuous variables.




```
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
dataset = pd.read_csv('breast_cancer_bd.csv')
sns.heatmap(dataset.corr())
plt.show()
```

Machine Learning Models:

1)Logistic Regression:

- Dataset is split between training and test dataset using inbuilt function.
- Logistic regression consists of two components: sigmoid function and feature with weights.

  Y= 1/(1+e^(-z))

- Z above is nothing but dot product of weights of x and weights w.
- We also have a loss function in our model that is given by formula below:

  f(y_actual,y_predicted) = [-y_actual*log(y_predicted)-(1-y_actual)*log(1-y_predicted)]

- We use this loss in the cost function of the model. Derivates are also calculated for weights and bias because it helps in the updating step of this algorithm. It tells us how much we should change the weights and in which direction should we go.
- This algorithm is used to classify the dataset in a class of whether the value or point is in the class or not in the class. This is used to solve some problems that are related to classification.
- Here like discussed above we have a dataset that has 458 benign and 241 malign values. Since these two values fall under two different classes we have implemented this algorithm to assign observations to a class. It has 699 samples.
- At first we start of by assigning weights and bias. Then we start off by performing forward propagation and updating values of weights and bias for 90 epochs.
- We then use the sigmoid function to calculate the cost which is also called logistic function. This function helps to map the value between 0 and 1. It takes the feature and weights as input and gives results between 0 and 1 as output.
- In the end it classifies data given by z to 1 if it is >0.5 and to 0 if it is less than 0.5 .
- This model is able to give accuracy of 94.72% on training data and 93.85% on the test data.

Output:

```
Accuracy of model built from scratch is:   93.71428571428571 %
Accuracy from inbuilt model using Sklearn library is:  94.85714285714286 %
Confusion Matrix for test data is :
[[105   7]
 [  4  59]]

Process finished with exit code 0
```

2) K Nearest Neighbours :

K Nearest Neighbours(KNN) is a supervised learning algorithm used for classification and regression predictive problems. Generally KNN is used for classification. The below two terms defines KNN in more detail

Lazy Learning algorithm : It is defined as a lazy learning algorithm because it doesn't learn anything in the training phase, it just loads the data in the training phase instead it learns in the testing phase.

Non-parametric learning algorithm : KNN is a non parametric algorithm as it doesn't consider any condition about the data.

KNN implementation in breast cancer detection : KNN is based on similarity features. Here we have two categories for classification i.e., 2 for Benign and 4 for Malignant. We want to classify what classes the sample of the training data set belongs to. So, by using KNN we will observe the behavior of the nearest sample from the test data set and classify the train set accordingly. We are dividing data into training and test data sets. Where training data is 75% and test data is 25%.

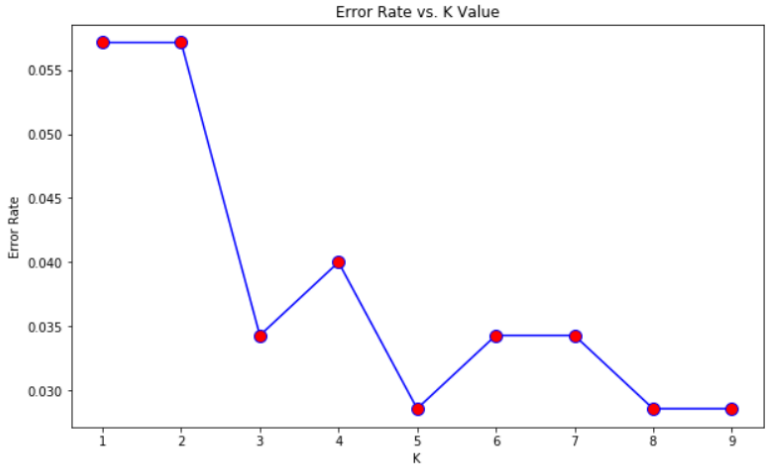
Now the question is how many neighbors to consider, i.e., selecting the value of K?

Value of K : When K = 1, it will overfit by performing well in the training phase but may not perform well in the testing phase. Another problem is that K is not set as even in binary classification. Let's say we have K = 6 and there is a tie between two labels from all 6 neighbor's

votes then there will be a confusion as to which class is to be assigned. Therefore setting the value of K as an odd number and square root of the total length of the dataset.

Value of K is calculated based on the minimum error. To calculate minimum error euclidean distance is found between training data set against test data set and prediction is calculated. Then the predicted labels are compared against test_label and mean of error is calculated. This process is repeated 10 times and from 10 iterations the least value of K is chosen to get the maximum accuracy. Below is the graph for error rate vs K for all 10 iterations and least error is for K = 5.

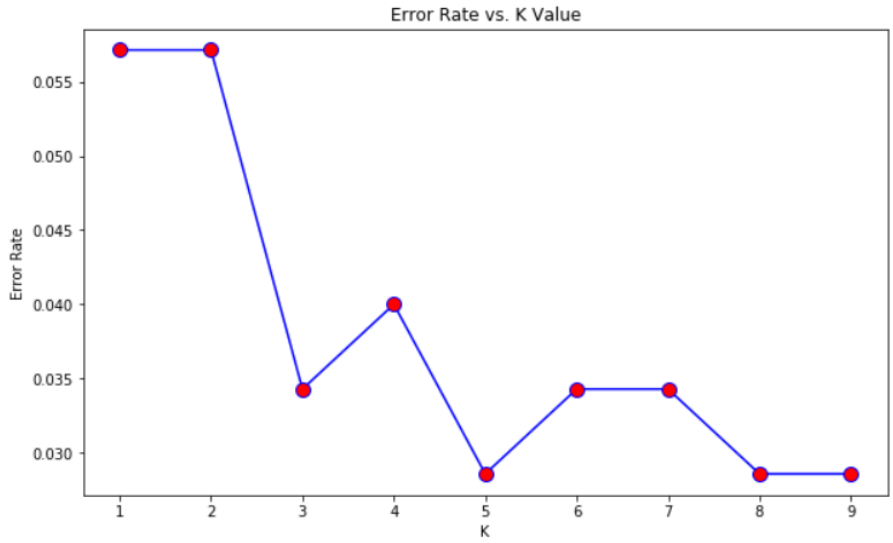
```
[1 1 1 ... 3 1 1]]
[0.05714285714285714, 0.05714285714285714, 0.03428571428571429, 0.04, 0.02857142857142857, 0.03428571428571429, 0.0342857142857
1429, 0.02857142857142857, 0.02857142857142857]
```



Now, we have K, training set and test set. From the same training set from which K is calculated , we can implement KNN for breast cancer detection and check its accuracy.

We calculated distances of the training set with respect to each sample in the test dataset using euclidean distances and calculated a distance list with the distances calculated and the class for training data set.

After finding distances and labels of the training set with respect to the test dataset , we found prediction labels by getting the majority of classes from the K nearest classes for the test data set. With the predicted label and actual test class label , we calculated confusion matrix, Accuracy , precision and recall for this model. Here is the final output.




```
5
Accuracy of KNN for K = 5 is 97.14285714285714%
confusion Matrix
[[107   3]
 [  2  63]]
Precision for Label 2 is 0.98
Recall for Label 2 is 0.97
Precision for Label 4 is 0.95
Recall for Label 4 is 0.97
```

Confusion Matrix : From the predicted labels for the test data set and actual labels from the test data set a confusion matrix can be built.

Now we have predicted the class for test data and actual test data to compare the accuracy .

Accuracy : There were a total 175 test samples, out of which it has detected the labels of 170 samples correctly. Therefore accuracy is (170/175)* 100 = 97.14 %. It means the classifier has detected only 5 samples incorrectly.

Recall : It shows that from all class label 2 , percentage of label 2 identified correctly. In this case the recall for Label 2 is 97%. Similarly from all class label 4, 97% of class 4 labels were identified correctly.

Precision : Precision shows the percentage of exact values(true positive) by all the values detected as true by the classifier (false positive + true positive). Here the precision for label 2 is 98% and precision for label 4 is 95%.

3) Naive Bayes Classifier:

A naive Bayes classifier is an algorithm that classifies things using Bayes theorem. Naive Bayes classifiers presume high, or naive, independence between data point properties. Spam filters, text analysis, and medical diagnosis are all common applications for naive Bayes classifiers. Because they are simple to implement, these classifiers are commonly employed in machine learning. It works with both continuous and discrete data and is extremely scalable in terms of predictors and data points. It is quick and can provide real-time forecasts.Here we are using Naive Bayes to indicate if a patient is at high risk for Breast cancer.The advantage of using Gaussian Naive Bayes to calculate accuracy is that it takes less time to train the data. In this experiment, we are implementing the Naive Bayes algorithm on the cancer dataset to diagnose breast cancer (In our experiments we will consider the label malignant as the positive label i.e,Given some characteristics of a tumor we want to identify if it is a malignant tumor or benign tumor). This dataset has 11 features, each with its own label column. The benign class distribution and malignant class are respectively 65.5% and 35% and labeled as 2 and 4 respectively. The entire dataset consists of 699 instances which were split into 25% test and 75% train data randomly.

We then trained the naive bayes classifier using the training data. We then used the classifier to make predictions on the test data. We have examined the confusion matrix for train and test data by which we determined the accuracy, precision, recall, f1 score and support. First, we note that the classifier has the accuracy of 95% on training and test dataset which generally indicates that the classifier is able to identify the positive and negative classes very well, however one keen observation here is that the dataset itself is unbalanced when it comes to positive and negative class distribution (65% benign, 35% malignant) For such datasets the more indicative matrix are generally precision, recall and f1 scores. At a high level, recall gives us the ability of a model to find all the relevant cases within a dataset. In this case, it gives us the classifiers ability to identify all the malignant cases correctly from the test dataset. Whereas precision is the fraction of relevant instances among the retrieved instances. In this case, out of all the instances that the classifier has identified as malignant, precision represents the fraction of true malignant cases in the test dataset. Finally, F1 score , which is a harmonic mean of precision and recall, shows an

aggregated metric of classifiers on the dataset.To conclude, we see that the classifier gives a precision of 92% and a recall of 95%, which shows that the classifier is showing good performance on positive relevant classes even on such an unbalanced dataset.

Conclusion:

After implementing the above three algorithms to detect Breast Cancer, We can now compare the accuracies. We can see that KNN has the highest accuracy i.e., 97.14 % followed by Naive Bayes with accuracy 95 %, followed by logistic regression 93.71%

The below graph compares accuracy of the three models.