SAVITRIBAI PHULE PUNE UNIVERSITY

A PRELIMINARY PROJECT REPORT ON

"Breast Cancer Classification on Breast Cancer Wisconsin Dataset Using Machine Learning"

SUBMITTED TOWARDS THE PARTIAL FULFILMENT OF THE REQUIREMENTS OF

LABORATORY PRACTICE - III (MACHINE LEARNING)

Academic Year: 2019-20

By:

Janhavi Khindkar(BECOA156)

Sohalee Kale(BECOA142)

Manthan Jetwan(BECOA139)

Under The Guidance of

Prof. Sushma Vispute

DEPARTMENT OF COMPUTER ENGINEERING,

PCET'S PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

Sector No. 26, Pradhikaran, Nigdi,
Pune - 411044

PCET'S PIMPRI CHINCHWAD COLLEGE OF ENGINEERING
Sector No. 26, Pradhikaran, Nigdi,
Pune - 411044

DEPARTMENT OF COMPUTER ENGINEERING

Certificate

This is to certify that the Mini Project report entitled

"Breast Cancer Classification on Breast Cancer Wisconsin Dataset Using Machine Learning"

Submitted By

Name of students	Roll no.
Janhavi Khindkar	BECOA156
Sohalee Kale	BECOA142
Manthan Jetwan	BECOA139

is approved by **Prof. Sushma Vispute** for submission. It is certified further that, to the best of my knowledge, the report represents work carried out by my students as the partial fulfillment for BE. Computer Engineering (Semester II) Laboratory-III Work (Machine Learning) as prescribed by the Savitribai Phule Pune University for the academic year 2019-20.

Prof. Sushma Vispute
(Mini Project Guide)

Place: Pune
Date:14 April 2020

Abstract

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 the medical field, where those methods are widely used in diagnosis and analysis to make decisions.

Keywords:

Breast Cancer ,Classification, Machine Learning, Models Selection,Feature Engineering, Standard Scaler ,Precision ,Recall , Confusion Matrix.

Table of Contents

Chapters	Chapter Content/Names	Page Number
Initial Pages	Title Page with Title of the topic, Name of the candidate with Exam Seat Number / Roll Number, Name of the Guide, Name of the Department, Institution and Year & University	
	Approval Sheet/Certificate	
	Abstract and Keywords	
	Table of Contents, List of Figures	
Chapter 1	**Introduction** 1.1 Problem Statement: 1.2 Project Idea 1.3 Motivation 1.4 Scope	**1**
Chapter 2	**Project Requirements** 2.1 H/W , S/W , resources, requirements & their detail explanation 2.2 Dataset Design 2.3 Hours Estimation	2-3
Chapter 3	**Module Description** 3.1 Block Diagram 3.2 Explanation	4-5
Chapter 4	**Results & Discussion**	6-10
Chapter 5	**Conclusion**	10

List of Figures

Figure number	Figure Name
1	Block diagram
2	Test train split
3.	Count of null entries
4.	Confusion Matrix
5.	Comparison of Algorithms Accuracy

Chapter 1

Introduction

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 the medical field, where those methods are widely used in diagnosis and analysis to make decisions.

1.1 Problem Statement

Breast Cancer Classification on Breast Cancer Wisconsin (Diagnostic) Data Set using Machine LearningThe goal of this project is to classify whether the breast cancer is benign or malignant by applying various machine learning techniques

1.2 Project Idea

The idea of this project is to apply data analysis and machine learning techniques to classify breast cancer and help in the treatment process.

1.3 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. Classification and data mining methods are an effective way to classify data. Especially in the medical field, where those methods are widely used in diagnosis and analysis to make decisions.

1.4 Scope

This project aims to observe which features are most helpful in predicting malignant or benign cancer and to see general trends that may aid us in model selection and hyperparameter selection. The goal is to classify whether the breast cancer is benign or malignant. To achieve this we have used machine learning classification methods to fit a function that can predict the discrete class of new input.

CHAPTER 2

2.PROJECT REQUIREMENTS

2.1 H/W , S/W , resources, requirements & their explanation:

S/W:

1.Jupyter Notebook:
JupyterLab is a web-based interactive development environment for Jupyter notebooks, code, and data. JupyterLab is flexible: configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning. JupyterLab is extensible and modular: write plugins that add new components and integrate with existing ones.

2.Sckit-Learn:
Scikit-learn is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.

3.Seaborn:
Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

4.Other python Libraries:
For visualization and mathematical operation libraries like numpy,pandas and matplotlib are used.

H/W:
1.Computer/Desktop

2.2 Dataset Design :

We will use the UCI Machine Learning Repository for breast cancer dataset.

The dataset used in this project is publicly available and was created by Dr. William H. Wolberg, physician at the University Of Wisconsin Hospital at Madison, Wisconsin, USA. To create the dataset Dr. Wolberg used fluid samples, taken from patients with solid breast masses and an easy-to-use graphical computer program called Xcyt, which is capable of performing the analysis of cytological features based on a digital scan. The program uses a curve-fitting algorithm, to compute ten features from each one of the cells in the sample, than it calculates the mean value, extreme value and standard error of each feature for the image, returning a 30 real-valued vector

Attribute Information:

1. ID number
2. Diagnosis (M = malignant, B = benign)
3. to 32 Ten real-valued features are computed for each cell nucleus:
1. radius (mean of distances from center to points on the perimeter)
2. texture (standard deviation of gray-scale values)
3. perimeter
4. area
5. smoothness (local variation in radius lengths)
6. compactness (perimeter² / area — 1.0)
7. concavity (severity of concave portions of the contour)
8. concave points (number of concave portions of the contour)
9. symmetry
10. fractal dimension (“coastline approximation” — 1)

2.3 Hours Estimation:

It would take a maximum 4 hours for the project to be ready and working.

CHAPTER 3
Module Description

3.1 BLOCK DIAGRAM:

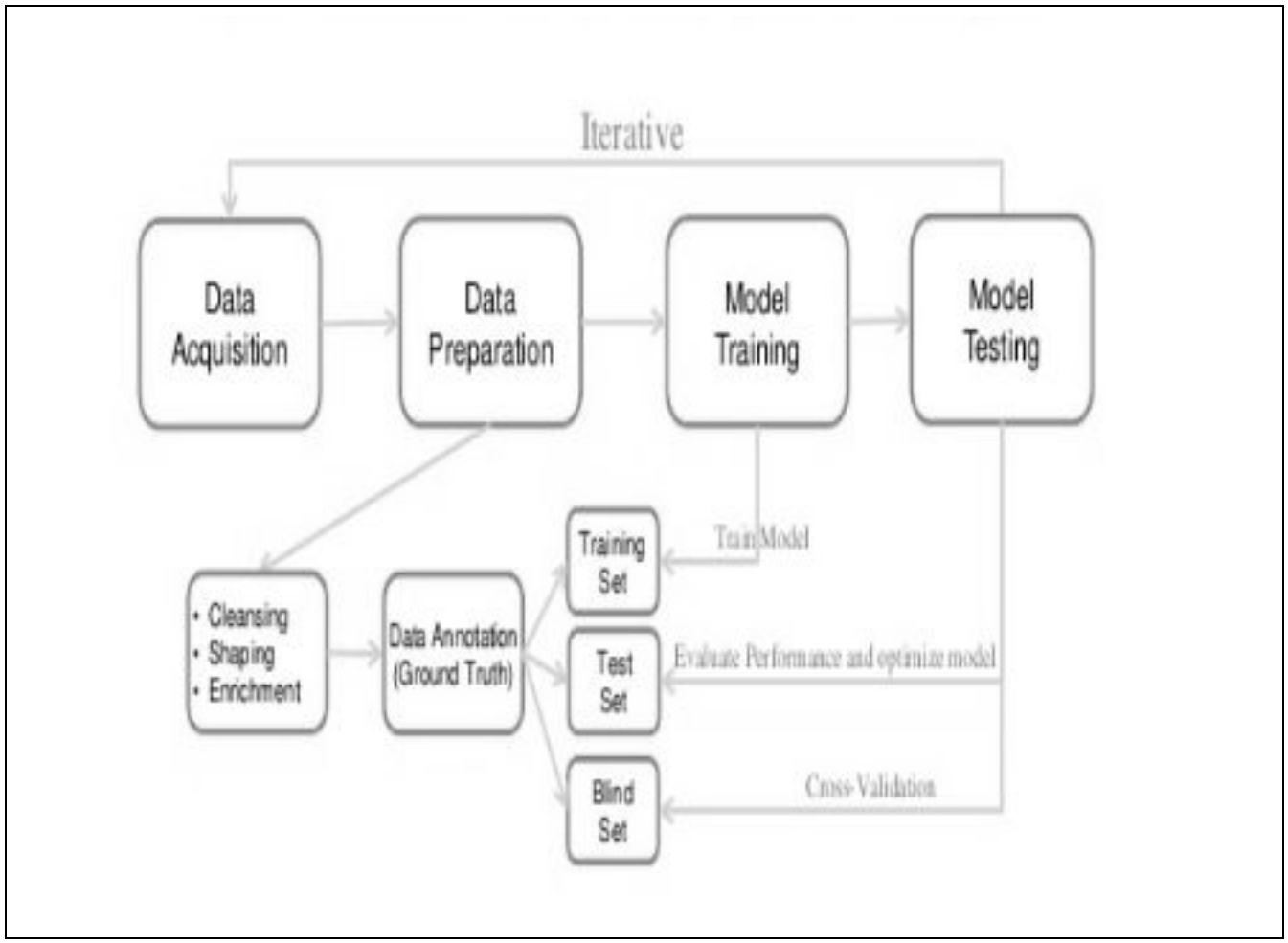

Fig.1 Block Diagram

Explanation:

1.Data Acquisition:

It is the process of collection of data regarding the respective problem statement.For our problem statement we will use the UCI Machine Learning Repository for breast cancer dataset.

2.Data Preparation:

Data preparation is the process of cleaning and transforming raw data prior to processing and analysis. It is an important step prior to processing and often involves reformatting data, making corrections to data and the combining of data sets to enrich data.Data preparation is often a lengthy undertaking for data professionals or business users, but it is essential as a prerequisite to put data in context in order to turn it into insights and eliminate bias resulting from poor data quality.
For example, the data preparation process usually includes standardizing data formats, enriching source data, and/or removing outliers.

3.Data splitting or Annotation:

The data we use is usually split into training data and test data. The training set contains a known output and the model learns on this data in order to be generalized to other data later on. We have the test dataset (or subset) in order to test our model's prediction on this subset.

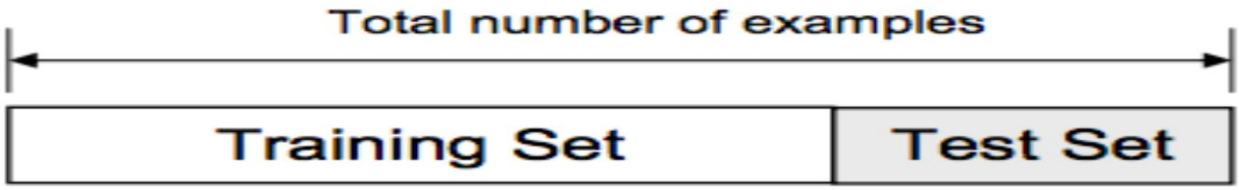

Fig. 2 Test Train Split

4.Model training:

Training a model simply means learning (determining) good values for all the weights and the bias from labeled examples. In supervised learning, a machine learning algorithm builds a model by examining many examples and attempting to find a model that minimizes loss; this process is called empirical risk minimization.

5.Model testing

The trained model is tested on a test dataset and is evaluated and accuray is found out.

CHAPTER 4

Results & Discussion

4.1 Source Code

Loading Data

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
#importing our cancer dataset
dataset = pd.read_csv('data.csv')
X = dataset.iloc[:, 2:32].values
Y = dataset.iloc[:, 1].values
```

Finding NULL data

```
dataset.isnull().sum()
dataset.isna().sum()
```

```
perimeter_mean                0
area_mean                     0
smoothness_mean               0
compactness_mean              0
concavity_mean                0
concave points_mean           0
symmetry_mean                 0
fractal_dimension_mean        0
radius_se                     0
texture_se                    0
perimeter_se                  0
area_se                       0
smoothness_se                 0
compactness_se                0
concavity_se                  0
concave points_se             0
symmetry_se                   0
fractal_dimension_se          0
radius_worst                  0
texture_worst                 0
perimeter_worst               0
area_worst                    0
smoothness_worst              0
compactness_worst             0
concavity_worst               0
concave points_worst          0
symmetry_worst                0
fractal_dimension_worst       0
Unnamed: 32                 569
dtype: int64
```

Fig 3 Count of Null Entries

Converting Categorical Data to numerical:

```
from sklearn.preprocessing import LabelEncoder
labelencoder_Y = LabelEncoder()
Y = labelencoder_Y.fit_transform(Y)
```

Splitting Dataset:

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.25,
random_state = 0)
```

Feature Scaling:

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Model Selection and Training on different Classifier:

```
#Using Logistic Regression Algorithm to the Training Set
from sklearn.linear_model import LogisticRegression
classifierLR = LogisticRegression(random_state = 0)
classifierLR.fit(X_train, Y_train)

#Using KNeighborsClassifier Method of neighbors class to use Nearest
Neighbor algorithm
from sklearn.neighbors import KNeighborsClassifier
classifierKNN = KNeighborsClassifier(n_neighbors = 5, metric =
'minkowski', p = 2)
classifierKNN.fit(X_train, Y_train)

#Using SVC method of svm class to use Support Vector Machine Algorithm
from sklearn.svm import SVC
classifierSVC = SVC(kernel = 'linear', random_state = 0)
classifierSVC.fit(X_train, Y_train)

#Using SVC method of svm class to use Kernel SVM Algorithm
from sklearn.svm import SVC
classifierSVCK = SVC(kernel = 'rbf', random_state = 0)
classifierSVCK.fit(X_train, Y_train)
```

```
#Using GaussianNB method of naïve_bayes class to use Naïve Bayes
Algorithm
from sklearn.naive_bayes import GaussianNB
classifierNB = GaussianNB()
classifierNB.fit(X_train, Y_train)

#Using DecisionTreeClassifier of tree class to use Decision Tree
Algorithm

from sklearn.tree import DecisionTreeClassifier
classifierDT = DecisionTreeClassifier(criterion = 'entropy', random_state
= 0)
classifierDT.fit(X_train, Y_train)

#Using RandomForestClassifier method of ensemble class to use Random
Forest Classification algorithm

from sklearn.ensemble import RandomForestClassifier
classifierRF = RandomForestClassifier(n_estimators = 10, criterion =
'entropy', random_state = 0)
classifierRF.fit(X_train, Y_train)

Y_pred = classifierSVC.predict(X_test)
```

SVM accuracy & Confusion matrix:

```
from sklearn.metrics import confusion_matrix, roc_curve, roc_auc_score
import seaborn as sns
cm = confusion_matrix(Y_test, Y_pred)
class_names=[0,1] # name  of classes
fig, ax = plt.subplots()
tick_marks = np.arange(len(class_names))
plt.xticks(tick_marks, class_names)
plt.yticks(tick_marks, class_names)
# create heatmap
sns.heatmap(pd.DataFrame(cm), annot=True, cmap="YlGnBu" ,fmt='g')
ax.xaxis.set_label_position("top")
plt.tight_layout()
plt.title('Confusion matrix', y=1.1)
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
```

```
tn, fp, fn, tp = cm.ravel()
precision = tp/(tp+fp) # 81/(81+23)
recall = tp/(tp+fn) # 81/(81+30)
print(precision, recall)
```

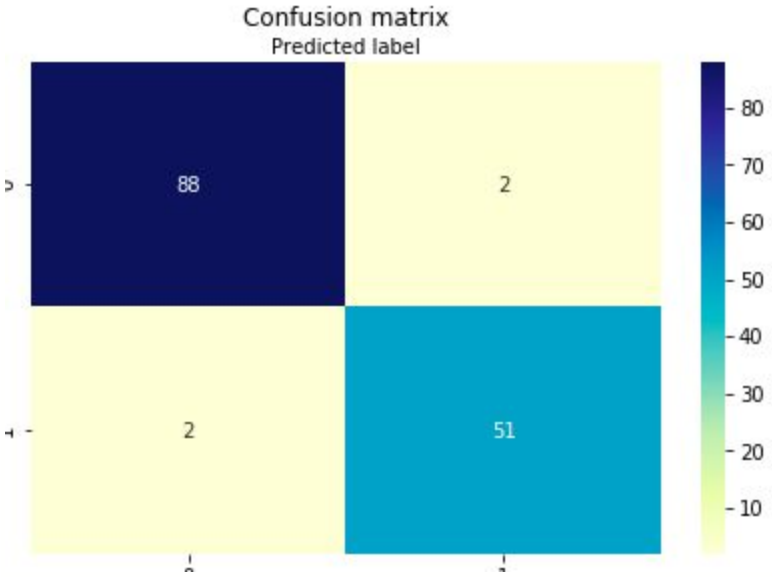

Fig 4 Confusion Matrix

Models Comparison & Cross Validation:

```
models = []
models.append(('LR', LogisticRegression()))
models.append(('KNN', KNeighborsClassifier()))
models.append(('CART', DecisionTreeClassifier()))
models.append(('NB', GaussianNB()))
models.append(('SVM', SVC()))
# evaluate each model in turn
results = []
names = []
scoring = 'accuracy'
for name, model in models:
      kfold = model_selection.KFold(n_splits=10, random_state=seed)
      cv_results = model_selection.cross_val_score(model, X, Y, cv=kfold,
scoring=scoring)
      results.append(cv_results)
      names.append(name)
      msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
      print(msg)
# boxplot algorithm comparison
```

```
fig = plt.figure()
fig.suptitle('Algorithm Comparison')
ax = fig.add_subplot(111)
plt.boxplot(results)
ax.set_xticklabels(names)
plt.show()
```

Results & Accuracy:

```
LR: 0.938534 (0.037788)
KNN: 0.926253 (0.046232)
CART: 0.931485 (0.028724)
NB: 0.936779 (0.036077)
SVM: 0.915758 (0.077166)
```

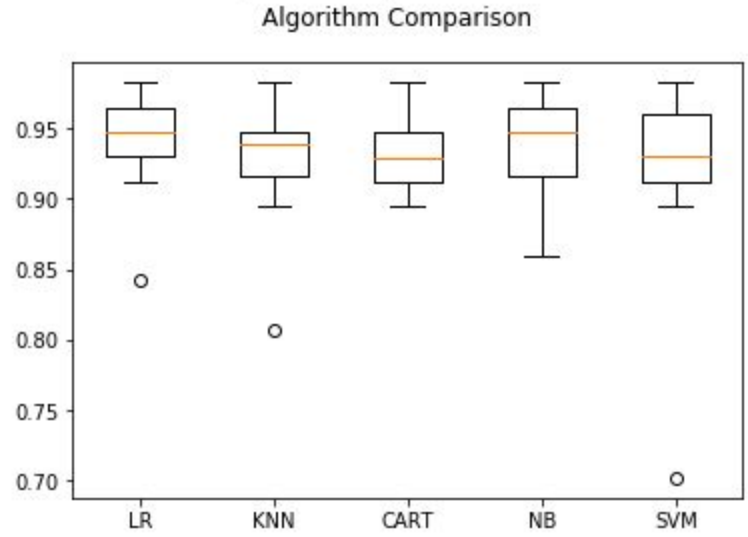

Fig 5 Algorithms Comparison

Conclusion:

So we have built our model based on various algorithms and compared their accuracies.Logistic Regression gives the best accuracy of all.Hence thus we have made a model to classify breast cancer.