**PART A**

(PART A : TO BE REFFERED BY STUDENTS)

**EXPERIMENT NO. 8**

**A.1 AIM: -** TO detect Breast Cancer Diagnosis via Logistic Regression

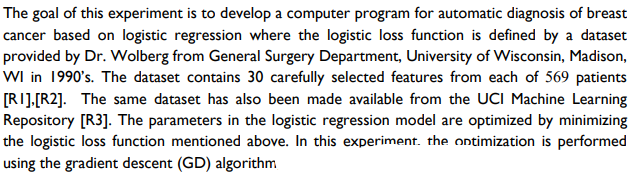
**A.2 Prerequisite**

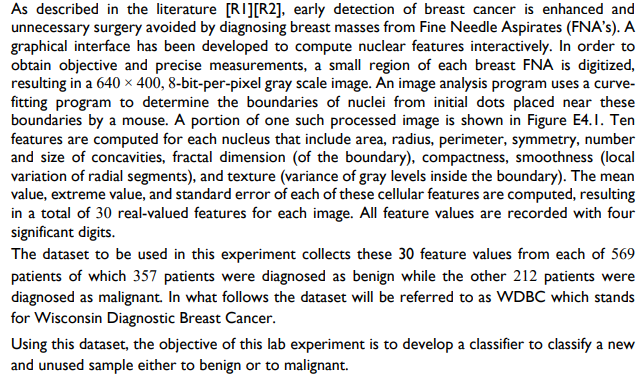
* Different programming language (Python or Java), Understanding of Machine Learning Algorithms, Machine Learning Algorithms

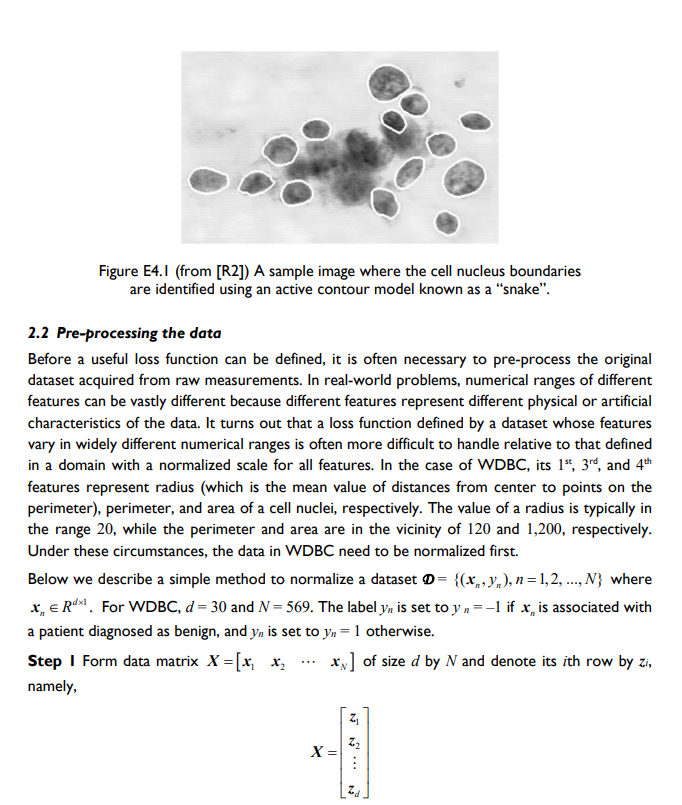
**A.3 Outcome**

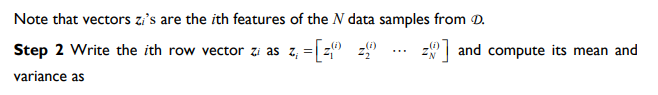
After successful completion of this experiment students will be able to Optimize the problem.

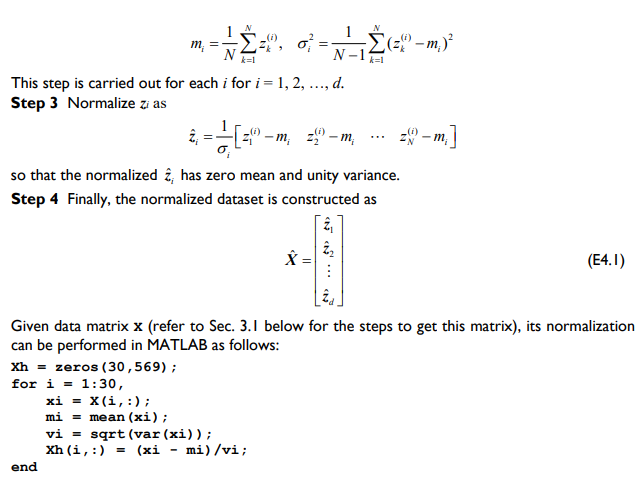
**A.4 Theory**



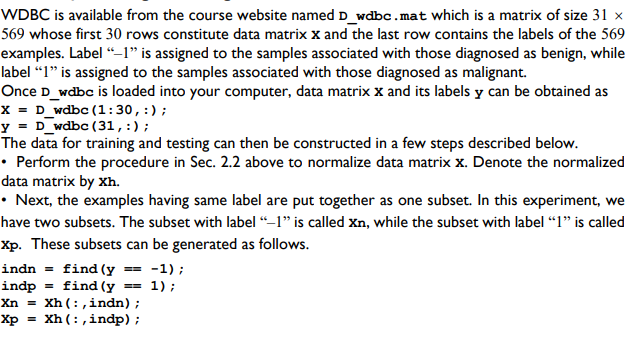


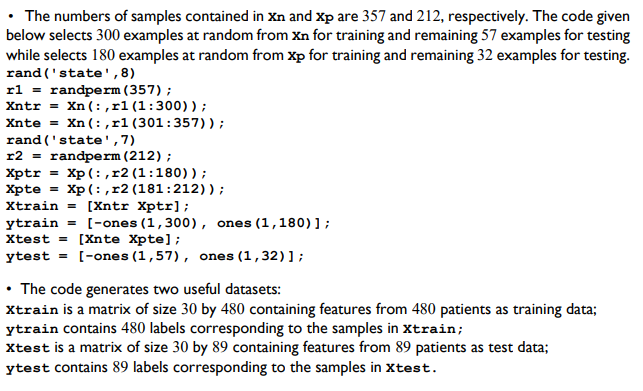


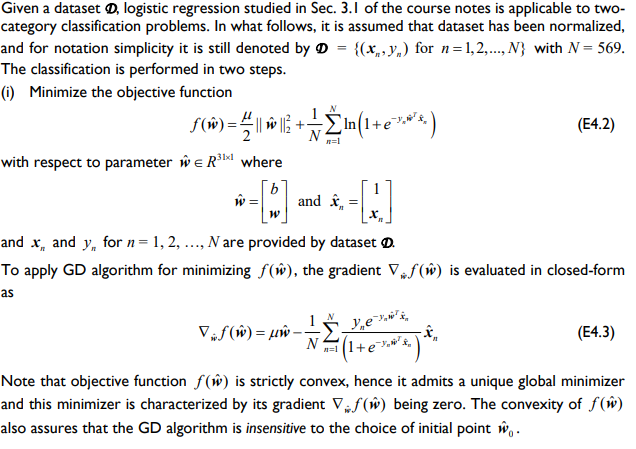


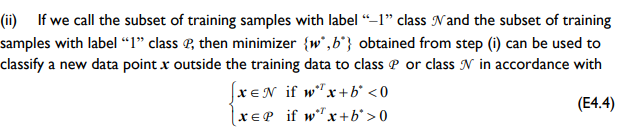


Procedure:









Links: <https://archive.ics.uci.edu/ml/datasets/breast%2Bcancer%2Bwisconsin%2B(Prognostic)>

Or

<https://www.kaggle.com/datasets/yasserhessein/dataset-breast-cancer-wisconsin>

PART B

(PART B : TO BE COMPLETED BY STUDENTS)

***(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Black board access available)***

|  |  |
| --- | --- |
| Roll No: C027 | Name: Vishesh Giyanani |
| Class : B | Batch : EB1 |
| Date of Experiment: 29/03/24 | Date of Submission |
| Grade : |  |

**B.1 Documentation written by student:**

import pandas as pd

from sklearn.model\_selection import train\_test\_split, GridSearchCV

from sklearn.linear\_model import LogisticRegression

from sklearn.preprocessing import StandardScaler

import numpy as np

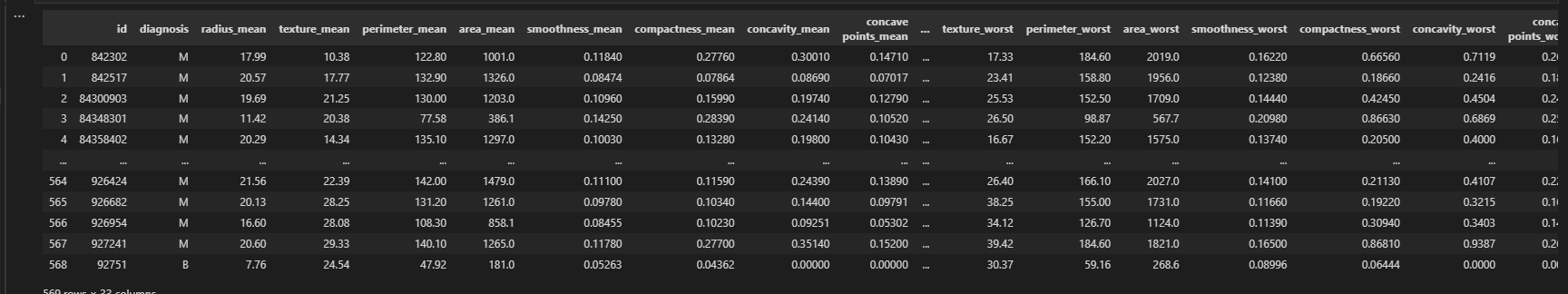
from sklearn.metrics import classification\_report, RocCurveDisplay

import warnings

warnings.filterwarnings("ignore")

data = pd.read\_csv('D:\\SEM X\\MLOA\\breast+cancer+wisconsin+prognostic\\data.csv')

data

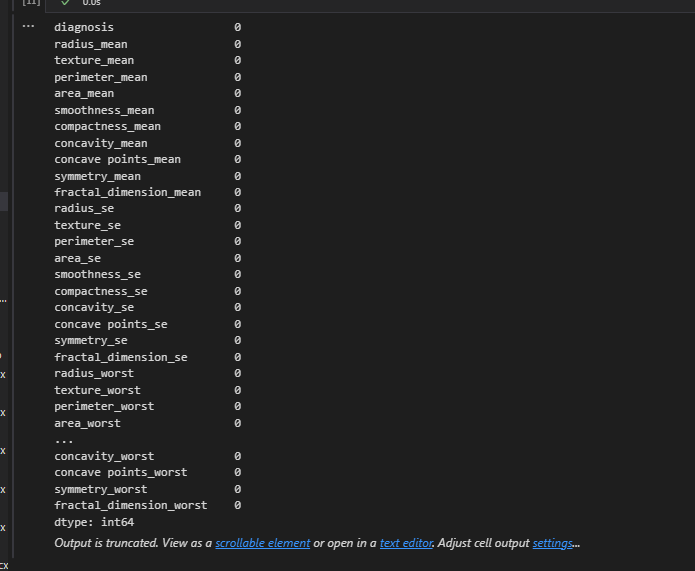


{"type":"dataframe","variable\_name":"data"}

data = data.drop(["id","Unnamed: 32"], axis=1)

scaler = StandardScaler()

data.isna().sum()



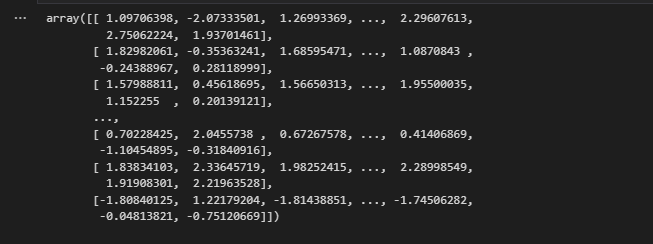
X = data.iloc[:,1:]

Y = data.iloc[:,0]

Y = np.where(Y == 'B', 0, 1)

X = scaler.fit\_transform(X)

X



X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,Y, random\_state = 42, test\_size= 0.2)

log = LogisticRegression()

grid = GridSearchCV(log, param\_grid= {

"penalty" : ["l1", "l2", "elasticnet"],

"C" : [0.3, 0.7, 1, 2],

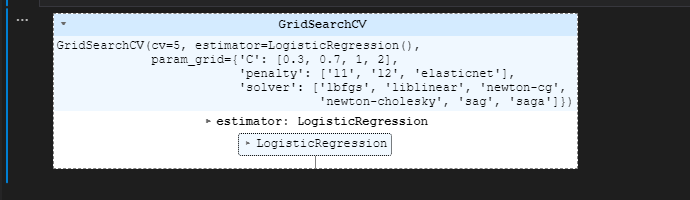
"solver" : ["lbfgs", "liblinear", "newton-cg",

"newton-cholesky", "sag", "saga"]

}, cv = 5

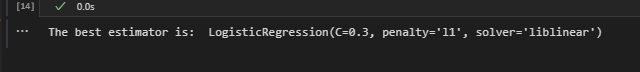
)

grid.fit(X\_train, y\_train )

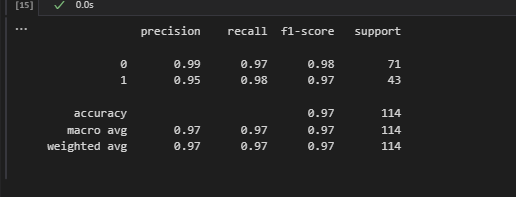


preds = grid.predict(X\_test)

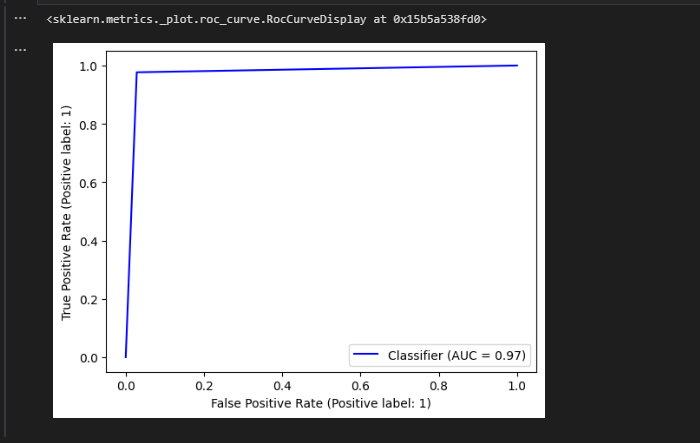
print("The best estimator is: " , grid.best\_estimator\_)



print(classification\_report(y\_test, preds))



RocCurveDisplay.from\_predictions(y\_test, preds, color='blue')



**B.2 Observations and learning:**

Implemented Logistic Regression on a Breast Cancer Dataset and observed the accuracy to be 0.97.

**B.3 Conclusion:**

Applied Logistic Regression on a Breast Cancer Dataset and the best fit models were found.