**ABSTRACT**

Breast cancer classification refers to the process of categorizing breast tumors or lesions into different classes based on their characteristics. The primary aim of classification is to distinguish between benign (non-cancerous) and malignant (cancerous) tumors. This classification plays a crucial role in breast cancer diagnosis and treatment planning.

# CHAPTER 1 INTRODUCTION

# Breast cancer is a significant health concern worldwide, affecting a large number of individuals, particularly women. Early detection and accurate classification of breast cancer are crucial for effective treatment and improved patient outcomes. In this report, we present a classification of breast cancer classification.

# The primary objective of this project is to develop a classification model capable of accurately distinguishing between malignant and benign breast tumours based on various clinical and pathological features. The classification model utilizes machine learning algorithms to learn patterns and relationships within the dataset and make predictions on new, unseen cases.

# The classification report aims to provide a comprehensive assessment of the performance of the developed model. It includes various evaluation metrics such as accuracy, precision, recall, F1-score, and specificity. These metrics provide insights into the model's overall performance, its ability to correctly identify malignant and benign cases, and the trade-off between false positives and false negatives.

# Additionally, the report will present a confusion matrix, which visually depicts the model's predictions compared to the actual ground truth. This matrix allows us to analyse the true positives, true negatives, false positives, and false negatives, providing a more detailed understanding of the model.

# CHAPTER 2 PROBLEM DEFINITION

Breast cancer is one of the most prevalent and life-threatening diseases affecting women worldwide. It is characterized by the uncontrolled growth of malignant cells in the breast tissue. Early detection and accurate diagnosis of breast cancer are critical for timely intervention and improved patient outcomes. However, identifying breast cancer and distinguishing between malignant and benign tumours can be challenging, even for experienced clinicians.

The problem at hand is to develop an effective breast cancer classification system that can accurately differentiate between malignant and benign cases. The classification system aims to leverage the power of machine learning algorithms and predictive modelling techniques to analyse various clinical and pathological features associated with breast cancer. By utilizing these features, the classification system can provide valuable insights to healthcare professionals, assisting in decision-making processes and guiding appropriate treatment plans.

The classification system seeks to address the following challenges:

1. Classification Accuracy: Developing a model that achieves high accuracy in distinguishing between malignant and benign breast tumours is of utmost importance. The classification system should minimize false positives (misclassifying benign cases as malignant) and false negatives (misclassifying malignant cases as benign) to ensure reliable predictions.

2. Early Detection: Early detection of breast cancer significantly improves the chances of successful treatment and patient survival. The classification system should be capable of identifying early-stage breast cancers with high accuracy, enabling timely intervention and reducing the risk of disease progression.

3. Feature Selection: Identifying the most relevant clinical and pathological features for accurate classification is crucial. The classification system should employ robust feature selection techniques to determine the optimal subset of features that contribute the most to the classification task. This helps in reducing computational complexity and potential overfitting, while ensuring the model captures the essential characteristics of breast cancer.

4. Generalization: The classification system should generalize well to unseen data, exhibiting consistent performance across different datasets and populations. It should be capable of handling variations in data quality, missing values, and diverse patient demographics, ensuring its applicability in real-world clinical settings.

Addressing these challenges and developing an accurate breast cancer classification system has the potential to revolutionize breast cancer diagnosis and treatment. It can facilitate early detection, improve patient outcomes, optimize resource allocation, and assist healthcare professionals in making informed decisions regarding personalized treatment plans. Moreover, a reliable classification system can contribute to ongoing research efforts, advancing our understanding of breast cancer and supporting the development of novel therapeutic interval.

# CHAPTER 3 LITERATURE REVIEW

Fong, S., & Nguyen, H. T. (2020). “A review on machine learning methods for breast cancer diagnosis and prediction”. Expert Systems with Applications, 115, 49-65.

This review provides an overview of various ML techniques employed in breast cancer classification, including support vector machines, random forests, artificial neural networks, and ensemble methods. It discusses the strengths, limitations, and performance of these methods, along with feature selection and pre-processing techniques commonly used in the field.

The research paper titled "Predicting Breast Cancer using Logistic Regression and Multi-Class Classifiers" explores the application of logistic regression and multi-class classifiers for breast cancer prediction.

The study aims to develop a predictive model for breast cancer using logistic regression and multi-class classifiers. The authors utilized the Wisconsin Diagnostic Breast Cancer (WBCD) dataset, which consists of various features extracted from digitized images of fine needle aspirates (FNA) of breast masses. The dataset includes benign and malignant cases.

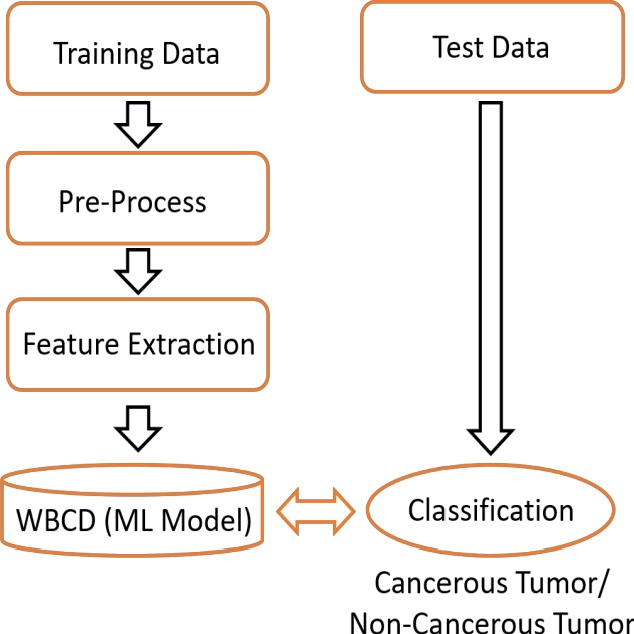
# The research paper titled “Detection and classification of breast cancer using logistic regression feature selection and GMDH classifier“

A deep investigation of previous works and a qualitative comparison among them. Using a new feature selection method based on logistic regression to reduce computational cost. Using a novel deep neural network, which can be tuned to increase accuracy or decrease computational cost. Using three different datasets to evaluate the proposed methods. Using a comprehensive set of evaluation metrics, including false positive rate. Reaching a high level of accuracy for all three datasets.

**CHAPTER 4 PROJECT DESCRIPTION**

Breast cancer is one of the most prevalent forms of cancer among women worldwide, emphasizing the importance of accurate and early diagnosis for effective treatment and improved patient outcomes. This project aims to develop a breast cancer classification system using machine learning techniques to aid in the accurate diagnosis and classification of breast tumours as either malignant or benign.

**CHAPTER 5 METHODOLOGY**



1. Data Collection: Relevant data related to breast cancer cases is collected, including clinical and pathological features such as patient demographics, tumour size, histological characteristics, genetic markers, and imaging data. The dataset should include labelled examples indicating whether each case is malignant or benign.

2. Data Pre-processing: The collected data is pre-processed to handle missing values, normalize features, and address any data inconsistencies. This step ensures the data is in a suitable format for further analysis.

3. Feature Selection and Engineering: Feature selection techniques are applied to identify the most informative features that contribute to breast cancer classification. This helps reduce dimensionality and focuses on the most relevant attributes. Additionally, new features may be engineered based on domain knowledge or feature transformations to enhance the classification performance.

4. Dataset Split: The dataset is split into training and testing sets. The training set is used to train the machine learning model, while the testing set is used to evaluate its performance on unseen data.

5. Model Evaluation: The trained model is evaluated using performance metrics such as accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC). These metrics assess the model's ability to correctly classify malignant and benign cases.

6. Model Validation: To ensure the model's generalization capability, accuracy score ,f1 score are employed. This involves repeatedly splitting the data into different training and validation subsets to assess the model's performance across multiple iterations.

7. Model Deployment: Once the model has been trained and validated, it can be deployed in real-world scenarios. This can involve integrating the model into existing clinical systems or developing a standalone software tool that accepts relevant patient data and provides breast cancer classification predictions.

8. Monitoring and Maintenance: The deployed model should be regularly monitored and updated as new data becomes available. This helps maintain its accuracy and adaptability to changing trends or patterns in breast cancer cases.

By leveraging machine learning algorithms and techniques, breast cancer classification models can aid healthcare professionals in accurate diagnosis, treatment planning, and decision-making, ultimately improving patient outcomes and contributing to advancements in breast cancer research.

# CHAPTER 6 EXPERIMENTATION

6.1: Dataset

The referenced dataset is called the Wisconsin Breast Cancer Diagnostic (WBCD) dataset, also known as the Wisconsin Diagnostic Breast Cancer (WDBC) dataset.

The WBCD dataset was created by Dr. William H. Wolberg at the University of Wisconsin Hospitals, Madison, in collaboration with the Wisconsin Breast Cancer Database team. It was obtained from fine-needle aspiration (FNA) samples of breast mass and contains features extracted from digitised images of those samples. The primary goal of this dataset is to predict whether a breast mass is benign.

(non-cancerous) or malignant (cancerous) based on the provided features

Here are the details of the WBCD dataset:

1. Number of Instances: 569
2. Number of Attributes: 32 (including the class attribute)
   * The first attribute is the unique patient ID number.
   * The second attribute is the diagnosis (M for malignant or B for benign).
   * The remaining 30 attributes represent various features computed from the FNA images. These features include radius, texture, smoothness, compactness, concavity, symmetry, fractal dimension, etc. Multiple measurements were taken for each feature, resulting in mean, standard error, and worst (largest) values.

6.2 Model creation

Logistic regression is used in our model

Logistic regression is a statistical modelling technique used for binary classification problems, where the goal is to predict a binary outcome or assign an observation to one of two possible classes. It is a type of regression analysis, but instead of predicting continuous values, it predicts the probability of an event occurring.

The main idea behind logistic regression is to model the relationship between the independent variables

(radius\_mean, texture\_mean,perimeter\_mean, area\_mean,smoothness\_mean) and the dependent variable (also called the target variable or output) using a logistic function, also known as the sigmoid function.

6.3 Model training

The dataset consists of 30 columns as features with 569 data entries and 1 column for prediction of the output corresponding to those features. The data is splatted into 2 parts i.e. training and testing data . The training data consists of 455 data entries with 30 columns and the test data contains 114 data entries with 30 columns.

The training set is used to train the model, while the validation set is used to evaluate its performance during training. Train the logistic regression model using the training set and calculate the accuracy score.

6.4 Testing and Performance Analysis

The trained model is evaluated using performance metrics such as accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC). These metrics assess the model's ability to correctly classify malignant and benign cases.

To ensure the model's generalization capability, accuracy score ,f1 score are employed. This involves repeatedly splitting the data into different training and validation subsets to assess the model's performance across multiple iterations.

# CHAPTER 7 RESULTS AND ANALYSIS

The below results are obtained by evaluation of the performance metrics

1.accuracy

2.precision

3.recall

4.f1 score

5. Confusion matrix

6.precision\_recall curve

Training data

Accuracy: 0.9472527472527472

Precision = 0.952054794520548

Recall = 0.9652777777777778

F1 Score = 0.9586206896551724

Test data;

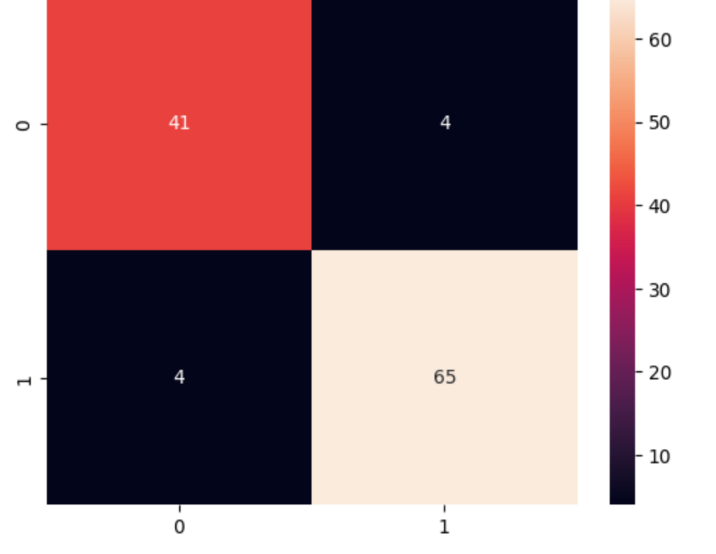
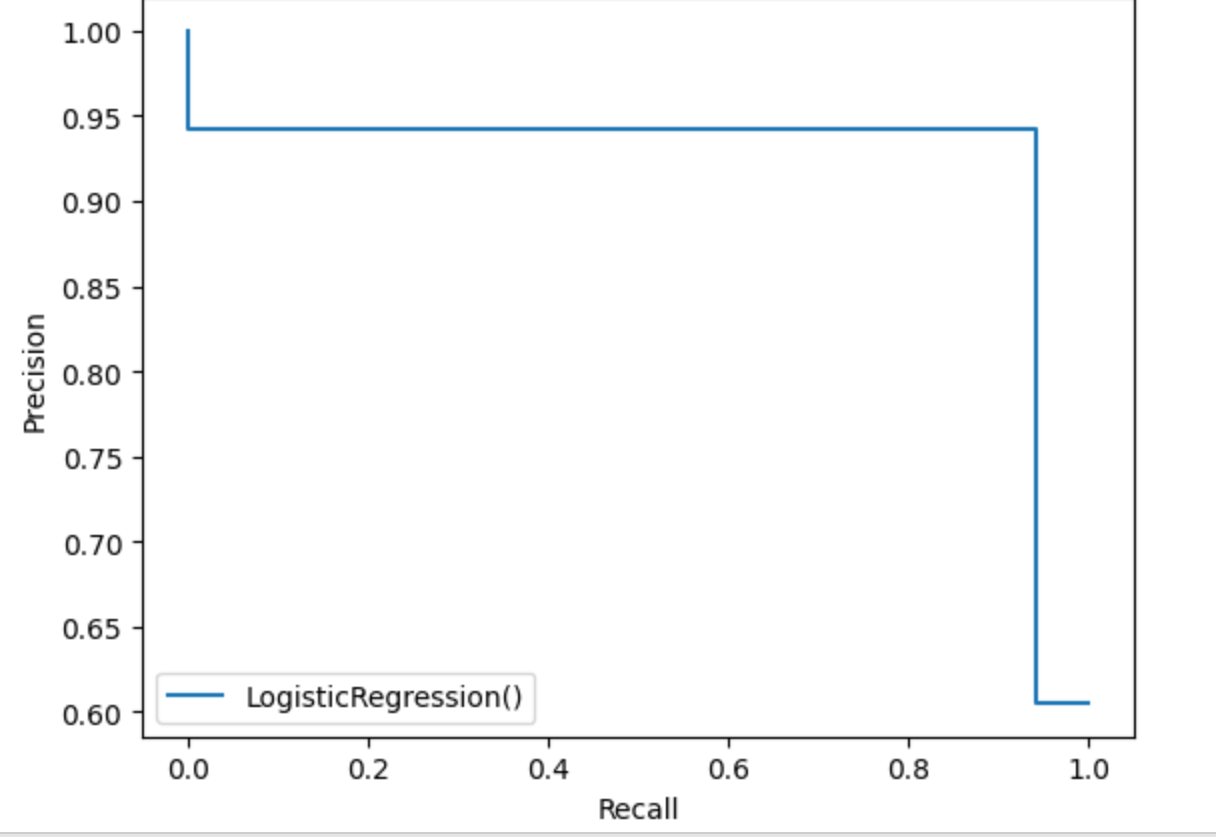
Accuracy: 0.9298245614035088

Precision = 0.9420289855072463

 Recall = 0.9420289855072463

F1 Score = 0.9420289855072463

Cross-validation score: [0.92763158 0.95394737 0.95364238]

**CHAPTER 8 CONCLUSION AND FUTURE WORK**

Breast cancer classification plays a crucial role in the diagnosis and treatment of breast cancer. With advancements in medical technology and research, various classification systems have been developed to categorize breast cancer based on different parameters, such as tumour size, lymph node involvement, hormone receptor status, HER2 status, and molecular subtypes.

Accurate classification of breast cancer is essential for determining the appropriate treatment strategy and predicting patient outcomes. The classification helps guide treatment decisions, such as surgery, chemotherapy, radiation therapy, targeted therapies, and hormone therapy. By identifying the specific characteristics of a breast cancer tumour, healthcare professionals can tailor treatments to individual patients, maximizing their chances of successful outcomes.

Moreover, breast cancer classification has contributed significantly to our understanding of the disease and its heterogeneity. The identification of different molecular subtypes, such as luminal A, luminal B, HER2-enriched, and triple-negative, has provided insights into the underlying biology of breast cancer and has led to the development of targeted therapies that specifically address the characteristics of each subtype.

Furthermore, classification systems aid in the standardization of research and clinical practice, allowing for consistent communication and comparison of results across different institutions and studies. This standardization enables researchers and healthcare providers to collaborate effectively, exchange knowledge, and advance our understanding of breast cancer.

# CHAPTER 9 REFERENCES

**AUTHOR**: Fong, S., & Nguyen, H. T

**TITLE:** A review on machine learning methods for breast cancer diagnosis and prediction. Expert Systems with Applications, 115, 49-65.

**LINK:** https://www.mdpi.com/2411-9660/2/2/13

**AUTHOR**: Marjan Naderan

**TITLE:** The research paper titled “Detection and classification of breast cancer using logistic regression feature selection and GMDH classifier**“**

**LINK:** https://www.sciencedirect.com/science/article/pii/S1532046420302173

**AUTHOR**: J. Sultana

**TITLE:** "Predicting Breast Cancer using Logistic Regression and Multi-Class Classifiers"

**LINK:**https://www.researchgate.net/publication/331233978\_Predicting\_Breast\_Cancer\_using\_Logistic\_Regression\_and\_Multi-Class\_Classifiers#:~:text=Logistic%20Regression%20method%20and%20Multi,data%20to%20build%20deep%20predictions.

https://www.kaggle.com/datasets/yasserh/breast-cancer-dataset%E2%80%8B

https://youtu.be/qGoCUpa8Qbo

https://www.sciencedirect.com/science/article/pii/S1532046420302173

https://www.researchgate.net/publication/331233978\_Predicting\_Breast\_Cancer\_using\_Logistic\_Regression\_and\_Multi-Class\_Classifiers#:~:text=Logistic%20Regression%20method%20and%20Multi,data%20to%20build%20deep%20predictions.

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2661033/

**PROGRAM/CODE**

Importing the Dependencies

import numpy as np

import pandas as pd

import sklearn.datasets

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

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score

from sklearn.model\_selection import cross\_val\_score

import warnings

warnings.filterwarnings('ignore')

from sklearn.naive\_bayes import MultinomialNB

Data Collection and Processing

# loading the data from sklearn

breast\_cancer\_dataset = sklearn.datasets.load\_breast\_cancer()

print(breast\_cancer\_dataset)

{'data': array([[1.799e+01, 1.038e+01, 1.228e+02, ..., 2.654e-01, 4.601e-01,

1.189e-01],

[2.057e+01, 1.777e+01, 1.329e+02, ..., 1.860e-01, 2.750e-01,

8.902e-02],

[1.969e+01, 2.125e+01, 1.300e+02, ..., 2.430e-01, 3.613e-01,

8.758e-02],

...,

[1.660e+01, 2.808e+01, 1.083e+02, ..., 1.418e-01, 2.218e-01,

7.820e-02],

[2.060e+01, 2.933e+01, 1.401e+02, ..., 2.650e-01, 4.087e-01,

1.240e-01],

[7.760e+00, 2.454e+01, 4.792e+01, ..., 0.000e+00, 2.871e-01,

7.039e-02]]), 'target': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,

0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0,

1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0,

1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1,

1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0,

0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,

1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1,

1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0,

0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0,

1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1,

1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0,

0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0,

0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0,

1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1,

1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1,

1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,

1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,

1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1,

1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1]), 'frame': None, 'target\_names': array(['malignant', 'benign'], dtype='<U9'), 'DESCR': '.. \_breast\_cancer\_dataset:\n\nBreast cancer wisconsin (diagnostic) dataset\n--------------------------------------------\n\n\*\*Data Set Characteristics:\*\*\n\n :Number of Instances: 569\n\n :Number of Attributes: 30 numeric, predictive attributes and the class\n\n :Attribute Information:\n - radius (mean of distances from center to points on the perimeter)\n - texture (standard deviation of gray-scale values)\n - perimeter\n - area\n - smoothness (local variation in radius lengths)\n - compactness (perimeter^2 / area - 1.0)\n - concavity (severity of concave portions of the contour)\n - concave points (number of concave portions of the contour)\n - symmetry\n - fractal dimension ("coastline approximation" - 1)\n\n The mean, standard error, and "worst" or largest (mean of the three\n worst/largest values) of these features were computed for each image,\n resulting in 30 features. For instance, field 0 is Mean Radius, field\n 10 is Radius SE, field 20 is Worst Radius.\n\n - class:\n - WDBC-Malignant\n - WDBC-Benign\n\n :Summary Statistics:\n\n ===================================== ====== ======\n Min Max\n ===================================== ====== ======\n radius (mean): 6.981 28.11\n texture (mean): 9.71 39.28\n perimeter (mean): 43.79 188.5\n area (mean): 143.5 2501.0\n smoothness (mean): 0.053 0.163\n compactness (mean): 0.019 0.345\n concavity (mean): 0.0 0.427\n concave points (mean): 0.0 0.201\n symmetry (mean): 0.106 0.304\n fractal dimension (mean): 0.05 0.097\n radius (standard error): 0.112 2.873\n texture (standard error): 0.36 4.885\n perimeter (standard error): 0.757 21.98\n area (standard error): 6.802 542.2\n smoothness (standard error): 0.002 0.031\n compactness (standard error): 0.002 0.135\n concavity (standard error): 0.0 0.396\n concave points (standard error): 0.0 0.053\n symmetry (standard error): 0.008 0.079\n fractal dimension (standard error): 0.001 0.03\n radius (worst): 7.93 36.04\n texture (worst): 12.02 49.54\n perimeter (worst): 50.41 251.2\n area (worst): 185.2 4254.0\n smoothness (worst): 0.071 0.223\n compactness (worst): 0.027 1.058\n concavity (worst): 0.0 1.252\n concave points (worst): 0.0 0.291\n symmetry (worst): 0.156 0.664\n fractal dimension (worst): 0.055 0.208\n ===================================== ====== ======\n\n :Missing Attribute Values: None\n\n :Class Distribution: 212 - Malignant, 357 - Benign\n\n :Creator: Dr. William H. Wolberg, W. Nick Street, Olvi L. Mangasarian\n\n :Donor: Nick Street\n\n :Date: November, 1995\n\nThis is a copy of UCI ML Breast Cancer Wisconsin (Diagnostic) datasets.\nhttps://goo.gl/U2Uwz2\n\nFeatures are computed from a digitized image of a fine needle\naspirate (FNA) of a breast mass. They describe\ncharacteristics of the cell nuclei present in the image.\n\nSeparating plane described above was obtained using\nMultisurface Method-Tree (MSM-T) [K. P. Bennett, "Decision Tree\nConstruction Via Linear Programming." Proceedings of the 4th\nMidwest Artificial Intelligence and Cognitive Science Society,\npp. 97-101, 1992], a classification method which uses linear\nprogramming to construct a decision tree. Relevant features\nwere selected using an exhaustive search in the space of 1-4\nfeatures and 1-3 separating planes.\n\nThe actual linear program used to obtain the separating plane\nin the 3-dimensional space is that described in:\n[K. P. Bennett and O. L. Mangasarian: "Robust Linear\nProgramming Discrimination of Two Linearly Inseparable Sets",\nOptimization Methods and Software 1, 1992, 23-34].\n\nThis database is also available through the UW CS ftp server:\n\nftp ftp.cs.wisc.edu\ncd math-prog/cpo-dataset/machine-learn/WDBC/\n\n.. topic:: References\n\n - W.N. Street, W.H. Wolberg and O.L. Mangasarian. Nuclear feature extraction \n for breast tumor diagnosis. IS&T/SPIE 1993 International Symposium on \n Electronic Imaging: Science and Technology, volume 1905, pages 861-870,\n San Jose, CA, 1993.\n - O.L. Mangasarian, W.N. Street and W.H. Wolberg. Breast cancer diagnosis and \n prognosis via linear programming. Operations Research, 43(4), pages 570-577, \n July-August 1995.\n - W.H. Wolberg, W.N. Street, and O.L. Mangasarian. Machine learning techniques\n to diagnose breast cancer from fine-needle aspirates. Cancer Letters 77 (1994) \n 163-171.', 'feature\_names': array(['mean radius', 'mean texture', 'mean perimeter', 'mean area',

'mean smoothness', 'mean compactness', 'mean concavity',

'mean concave points', 'mean symmetry', 'mean fractal dimension',

'radius error', 'texture error', 'perimeter error', 'area error',

'smoothness error', 'compactness error', 'concavity error',

'concave points error', 'symmetry error',

'fractal dimension error', 'worst radius', 'worst texture',

'worst perimeter', 'worst area', 'worst smoothness',

'worst compactness', 'worst concavity', 'worst concave points',

'worst symmetry', 'worst fractal dimension'], dtype='<U23'), 'filename': 'breast\_cancer.csv', 'data\_module': 'sklearn.datasets.data'}

# loading the data to a data frame

data\_frame = pd.DataFrame(breast\_cancer\_dataset.data, columns = breast\_cancer\_dataset.feature\_names)

# print the first 5 rows of the dataframe

data\_frame.head()

|  | **mean radius** | **mean texture** | **mean perimeter** | **mean area** | **mean smoothness** | **mean compactness** | **mean concavity** | **mean concave points** | **mean symmetry** | **mean fractal dimension** | **...** | **worst radius** | **worst texture** | **worst perimeter** | **worst area** | **worst smoothness** | **worst compactness** | **worst concavity** | **worst concave points** | **worst symmetry** | **worst fractal dimension** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 17.99 | 10.38 | 122.80 | 1001.0 | 0.11840 | 0.27760 | 0.3001 | 0.14710 | 0.2419 | 0.07871 | ... | 25.38 | 17.33 | 184.60 | 2019.0 | 0.1622 | 0.6656 | 0.7119 | 0.2654 | 0.4601 | 0.11890 |
| **1** | 20.57 | 17.77 | 132.90 | 1326.0 | 0.08474 | 0.07864 | 0.0869 | 0.07017 | 0.1812 | 0.05667 | ... | 24.99 | 23.41 | 158.80 | 1956.0 | 0.1238 | 0.1866 | 0.2416 | 0.1860 | 0.2750 | 0.08902 |
| **2** | 19.69 | 21.25 | 130.00 | 1203.0 | 0.10960 | 0.15990 | 0.1974 | 0.12790 | 0.2069 | 0.05999 | ... | 23.57 | 25.53 | 152.50 | 1709.0 | 0.1444 | 0.4245 | 0.4504 | 0.2430 | 0.3613 | 0.08758 |
| **3** | 11.42 | 20.38 | 77.58 | 386.1 | 0.14250 | 0.28390 | 0.2414 | 0.10520 | 0.2597 | 0.09744 | ... | 14.91 | 26.50 | 98.87 | 567.7 | 0.2098 | 0.8663 | 0.6869 | 0.2575 | 0.6638 | 0.17300 |
| **4** | 20.29 | 14.34 | 135.10 | 1297.0 | 0.10030 | 0.13280 | 0.1980 | 0.10430 | 0.1809 | 0.05883 | ... | 22.54 | 16.67 | 152.20 | 1575.0 | 0.1374 | 0.2050 | 0.4000 | 0.1625 | 0.2364 | 0.07678 |

5 rows × 30 columns

# adding the 'target' column to the data frame

data\_frame['label'] = breast\_cancer\_dataset.target

# print last 5 rows of the dataframe

data\_frame.tail()

|  | **mean radius** | **mean texture** | **mean perimeter** | **mean area** | **mean smoothness** | **mean compactness** | **mean concavity** | **mean concave points** | **mean symmetry** | **mean fractal dimension** | **...** | **worst texture** | **worst perimeter** | **worst area** | **worst smoothness** | **worst compactness** | **worst concavity** | **worst concave points** | **worst symmetry** | **worst fractal dimension** | **label** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **564** | 21.56 | 22.39 | 142.00 | 1479.0 | 0.11100 | 0.11590 | 0.24390 | 0.13890 | 0.1726 | 0.05623 | ... | 26.40 | 166.10 | 2027.0 | 0.14100 | 0.21130 | 0.4107 | 0.2216 | 0.2060 | 0.07115 | 0 |
| **565** | 20.13 | 28.25 | 131.20 | 1261.0 | 0.09780 | 0.10340 | 0.14400 | 0.09791 | 0.1752 | 0.05533 | ... | 38.25 | 155.00 | 1731.0 | 0.11660 | 0.19220 | 0.3215 | 0.1628 | 0.2572 | 0.06637 | 0 |
| **566** | 16.60 | 28.08 | 108.30 | 858.1 | 0.08455 | 0.10230 | 0.09251 | 0.05302 | 0.1590 | 0.05648 | ... | 34.12 | 126.70 | 1124.0 | 0.11390 | 0.30940 | 0.3403 | 0.1418 | 0.2218 | 0.07820 | 0 |
| **567** | 20.60 | 29.33 | 140.10 | 1265.0 | 0.11780 | 0.27700 | 0.35140 | 0.15200 | 0.2397 | 0.07016 | ... | 39.42 | 184.60 | 1821.0 | 0.16500 | 0.86810 | 0.9387 | 0.2650 | 0.4087 | 0.12400 | 0 |
| **568** | 7.76 | 24.54 | 47.92 | 181.0 | 0.05263 | 0.04362 | 0.00000 | 0.00000 | 0.1587 | 0.05884 | ... | 30.37 | 59.16 | 268.6 | 0.08996 | 0.06444 | 0.0000 | 0.0000 | 0.2871 | 0.07039 | 1 |

5 rows × 31 columns

# number of rows and columns in the dataset

data\_frame.shape

(569, 31)

# getting some information about the data

data\_frame.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 569 entries, 0 to 568

Data columns (total 31 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 mean radius 569 non-null float64

1 mean texture 569 non-null float64

2 mean perimeter 569 non-null float64

3 mean area 569 non-null float64

4 mean smoothness 569 non-null float64

5 mean compactness 569 non-null float64

6 mean concavity 569 non-null float64

7 mean concave points 569 non-null float64

8 mean symmetry 569 non-null float64

9 mean fractal dimension 569 non-null float64

10 radius error 569 non-null float64

11 texture error 569 non-null float64

12 perimeter error 569 non-null float64

13 area error 569 non-null float64

14 smoothness error 569 non-null float64

15 compactness error 569 non-null float64

16 concavity error 569 non-null float64

17 concave points error 569 non-null float64

18 symmetry error 569 non-null float64

19 fractal dimension error 569 non-null float64

20 worst radius 569 non-null float64

21 worst texture 569 non-null float64

22 worst perimeter 569 non-null float64

23 worst area 569 non-null float64

24 worst smoothness 569 non-null float64

25 worst compactness 569 non-null float64

26 worst concavity 569 non-null float64

27 worst concave points 569 non-null float64

28 worst symmetry 569 non-null float64

29 worst fractal dimension 569 non-null float64

30 label 569 non-null int64

dtypes: float64(30), int64(1)

memory usage: 137.9 KB

# checking for missing values

data\_frame.isnull().sum()

mean radius 0 mean texture 0 mean perimeter 0 mean area 0 mean smoothness 0 mean compactness 0 mean concavity 0 mean concave points 0 mean symmetry 0 mean fractal dimension 0 radius error 0 texture error 0 perimeter error 0 area error 0 smoothness error 0 compactness error 0 concavity error 0 concave points error 0 symmetry error 0 fractal dimension error 0 worst radius 0 worst texture 0 worst perimeter 0 worst area 0 worst smoothness 0 worst compactness 0 worst concavity 0 worst concave points 0 worst symmetry 0 worst fractal dimension 0 label 0 dtype: int64

# statistical measures about the data

data\_frame.describe()

|  | **mean radius** | **mean texture** | **mean perimeter** | **mean area** | **mean smoothness** | **mean compactness** | **mean concavity** | **mean concave points** | **mean symmetry** | **mean fractal dimension** | **...** | **worst texture** | **worst perimeter** | **worst area** | **worst smoothness** | **worst compactness** | **worst concavity** | **worst concave points** | **worst symmetry** | **worst fractal dimension** | **label** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **count** | 569.000000 | 569.000000 | 569.000000 | 569.000000 | 569.000000 | 569.000000 | 569.000000 | 569.000000 | 569.000000 | 569.000000 | ... | 569.000000 | 569.000000 | 569.000000 | 569.000000 | 569.000000 | 569.000000 | 569.000000 | 569.000000 | 569.000000 | 569.000000 |
| **mean** | 14.127292 | 19.289649 | 91.969033 | 654.889104 | 0.096360 | 0.104341 | 0.088799 | 0.048919 | 0.181162 | 0.062798 | ... | 25.677223 | 107.261213 | 880.583128 | 0.132369 | 0.254265 | 0.272188 | 0.114606 | 0.290076 | 0.083946 | 0.627417 |
| **std** | 3.524049 | 4.301036 | 24.298981 | 351.914129 | 0.014064 | 0.052813 | 0.079720 | 0.038803 | 0.027414 | 0.007060 | ... | 6.146258 | 33.602542 | 569.356993 | 0.022832 | 0.157336 | 0.208624 | 0.065732 | 0.061867 | 0.018061 | 0.483918 |
| **min** | 6.981000 | 9.710000 | 43.790000 | 143.500000 | 0.052630 | 0.019380 | 0.000000 | 0.000000 | 0.106000 | 0.049960 | ... | 12.020000 | 50.410000 | 185.200000 | 0.071170 | 0.027290 | 0.000000 | 0.000000 | 0.156500 | 0.055040 | 0.000000 |
| **25%** | 11.700000 | 16.170000 | 75.170000 | 420.300000 | 0.086370 | 0.064920 | 0.029560 | 0.020310 | 0.161900 | 0.057700 | ... | 21.080000 | 84.110000 | 515.300000 | 0.116600 | 0.147200 | 0.114500 | 0.064930 | 0.250400 | 0.071460 | 0.000000 |
| **50%** | 13.370000 | 18.840000 | 86.240000 | 551.100000 | 0.095870 | 0.092630 | 0.061540 | 0.033500 | 0.179200 | 0.061540 | ... | 25.410000 | 97.660000 | 686.500000 | 0.131300 | 0.211900 | 0.226700 | 0.099930 | 0.282200 | 0.080040 | 1.000000 |
| **75%** | 15.780000 | 21.800000 | 104.100000 | 782.700000 | 0.105300 | 0.130400 | 0.130700 | 0.074000 | 0.195700 | 0.066120 | ... | 29.720000 | 125.400000 | 1084.000000 | 0.146000 | 0.339100 | 0.382900 | 0.161400 | 0.317900 | 0.092080 | 1.000000 |
| **max** | 28.110000 | 39.280000 | 188.500000 | 2501.000000 | 0.163400 | 0.345400 | 0.426800 | 0.201200 | 0.304000 | 0.097440 | ... | 49.540000 | 251.200000 | 4254.000000 | 0.222600 | 1.058000 | 1.252000 | 0.291000 | 0.663800 | 0.207500 | 1.000000 |

8 rows × 31 columns

# checking the distribution of Target Varibale

data\_frame['label'].value\_counts()

1 357 0 212 Name: label, dtype: int64

#1 --> Benign

0 --> Malignant

data\_frame.groupby('label').mean()

|  | **mean radius** | **mean texture** | **mean perimeter** | **mean area** | **mean smoothness** | **mean compactness** | **mean concavity** | **mean concave points** | **mean symmetry** | **mean fractal dimension** | **...** | **worst radius** | **worst texture** | **worst perimeter** | **worst area** | **worst smoothness** | **worst compactness** | **worst concavity** | **worst concave points** | **worst symmetry** | **worst fractal dimension** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **label** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **0** | 17.462830 | 21.604906 | 115.365377 | 978.376415 | 0.102898 | 0.145188 | 0.160775 | 0.087990 | 0.192909 | 0.062680 | ... | 21.134811 | 29.318208 | 141.370330 | 1422.286321 | 0.144845 | 0.374824 | 0.450606 | 0.182237 | 0.323468 | 0.091530 |
| **1** | 12.146524 | 17.914762 | 78.075406 | 462.790196 | 0.092478 | 0.080085 | 0.046058 | 0.025717 | 0.174186 | 0.062867 | ... | 13.379801 | 23.515070 | 87.005938 | 558.899440 | 0.124959 | 0.182673 | 0.166238 | 0.074444 | 0.270246 | 0.079442 |

2 rows × 30 columns

**Separating the features and target**

X = data\_frame.drop(columns='label', axis=1)

Y = data\_frame['label']

print(X)

mean radius mean texture mean perimeter mean area mean smoothness \

0 17.99 10.38 122.80 1001.0 0.11840

1 20.57 17.77 132.90 1326.0 0.08474

2 19.69 21.25 130.00 1203.0 0.10960

3 11.42 20.38 77.58 386.1 0.14250

4 20.29 14.34 135.10 1297.0 0.10030

.. ... ... ... ... ...

564 21.56 22.39 142.00 1479.0 0.11100

565 20.13 28.25 131.20 1261.0 0.09780

566 16.60 28.08 108.30 858.1 0.08455

567 20.60 29.33 140.10 1265.0 0.11780

568 7.76 24.54 47.92 181.0 0.05263

mean compactness mean concavity mean concave points mean symmetry \

0 0.27760 0.30010 0.14710 0.2419

1 0.07864 0.08690 0.07017 0.1812

2 0.15990 0.19740 0.12790 0.2069

3 0.28390 0.24140 0.10520 0.2597

4 0.13280 0.19800 0.10430 0.1809

.. ... ... ... ...

564 0.11590 0.24390 0.13890 0.1726

565 0.10340 0.14400 0.09791 0.1752

566 0.10230 0.09251 0.05302 0.1590

567 0.27700 0.35140 0.15200 0.2397

568 0.04362 0.00000 0.00000 0.1587

mean fractal dimension ... worst radius worst texture \

0 0.07871 ... 25.380 17.33

1 0.05667 ... 24.990 23.41

2 0.05999 ... 23.570 25.53

3 0.09744 ... 14.910 26.50

4 0.05883 ... 22.540 16.67

.. ... ... ... ...

564 0.05623 ... 25.450 26.40

565 0.05533 ... 23.690 38.25

566 0.05648 ... 18.980 34.12

567 0.07016 ... 25.740 39.42

568 0.05884 ... 9.456 30.37

worst perimeter worst area worst smoothness worst compactness \

0 184.60 2019.0 0.16220 0.66560

1 158.80 1956.0 0.12380 0.18660

2 152.50 1709.0 0.14440 0.42450

3 98.87 567.7 0.20980 0.86630

4 152.20 1575.0 0.13740 0.20500

.. ... ... ... ...

564 166.10 2027.0 0.14100 0.21130

565 155.00 1731.0 0.11660 0.19220

566 126.70 1124.0 0.11390 0.30940

567 184.60 1821.0 0.16500 0.86810

568 59.16 268.6 0.08996 0.06444

worst concavity worst concave points worst symmetry \

0 0.7119 0.2654 0.4601

1 0.2416 0.1860 0.2750

2 0.4504 0.2430 0.3613

3 0.6869 0.2575 0.6638

4 0.4000 0.1625 0.2364

.. ... ... ...

564 0.4107 0.2216 0.2060

565 0.3215 0.1628 0.2572

566 0.3403 0.1418 0.2218

567 0.9387 0.2650 0.4087

568 0.0000 0.0000 0.2871

worst fractal dimension

0 0.11890

1 0.08902

2 0.08758

3 0.17300

4 0.07678

.. ...

564 0.07115

565 0.06637

566 0.07820

567 0.12400

568 0.07039

[569 rows x 30 columns]

print(Y)

0 0

1 0

2 0

3 0

4 0

..

564 0

565 0

566 0

567 0

568 1

Name: label, Length: 569, dtype: int64

**Splitting the data into training data & Testing data**

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.2, random\_state=2)

print(X.shape, X\_train.shape, X\_test.shape)

(569, 30) (455, 30) (114, 30)

**Model Training**

**Logistic Regression**

model = LogisticRegression()

# training the Logistic Regression model using Training data

model.fit(X\_train, Y\_train)

**Model Evaluation**

**Accuracy Score**

# accuracy on training data

X\_train\_prediction = model.predict(X\_train)

training\_data\_accuracy = accuracy\_score(Y\_train, X\_train\_prediction)

print('Accuracy on training data = ', training\_data\_accuracy)

Accuracy on training data = 0.9472527472527472

# accuracy on test data

X\_test\_prediction = model.predict(X\_test)

test\_data\_accuracy = accuracy\_score(Y\_test, X\_test\_prediction)

print('Accuracy on test data = ', test\_data\_accuracy)

Accuracy on training data = 0.9472527472527472

**Building a Predictive System**

input\_data = (13.54,14.36,87.46,566.3,0.09779,0.08129,0.06664,0.04781,0.1885,0.05766,0.2699,0.7886,2.058,23.56,0.008462,0.0146,0.02387,0.01315,0.0198,0.0023,15.11,19.26,99.7,711.2,0.144,0.1773,0.239,0.1288,0.2977,0.07259)

# change the input data to a numpy array

input\_data\_as\_numpy\_array = np.asarray(input\_data)

# reshape the numpy array as we are predicting for one datapoint

input\_data\_reshaped = input\_data\_as\_numpy\_array.reshape(1,-1)

prediction = model.predict(input\_data\_reshaped)

print(prediction)

if (prediction[0] == 0):

  print('The Breast cancer is Malignant')

else:

  print('The Breast Cancer is Benign')

[1]

The Breast Cancer is Benign

from sklearn.metrics import precision\_score

from sklearn.metrics import recall\_score

from sklearn.metrics import f1\_score

# precision for training data predictions

precision\_train = precision\_score(Y\_train, X\_train\_prediction)

print('Training data Precision =', precision\_train)

Training data Precision = 0.952054794520548

# precision for test data predictions

precision\_test = precision\_score(Y\_test, X\_test\_prediction)

print('Test data Precision =', precision\_test)

Test data Precision = 0.9420289855072463

def precision\_recall\_f1\_score(true\_labels, pred\_labels):

  precision\_value = precision\_score(true\_labels, pred\_labels)

  recall\_value = recall\_score(true\_labels, pred\_labels)

  f1\_score\_value = f1\_score(true\_labels, pred\_labels)

  print('Precision =',precision\_value)

  print('Recall =',recall\_value)

  print('F1 Score =',f1\_score\_value)

# classification metrics for training data

precision\_recall\_f1\_score(Y\_train, X\_train\_prediction)

Precision = 0.952054794520548

Recall = 0.9652777777777778

F1 Score = 0.9586206896551724

# classification metrics for test data

precision\_recall\_f1\_score(Y\_test, X\_test\_prediction)

Precision = 0.9420289855072463

Recall = 0.9420289855072463

F1 Score = 0.9420289855072463

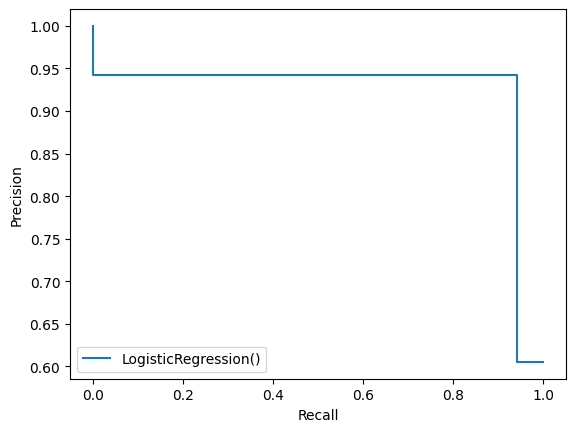
from sklearn.metrics import precision\_recall\_curve

from sklearn.metrics import PrecisionRecallDisplay

y\_pred = model.predict(X\_test)

prec, recall, \_ = precision\_recall\_curve(Y\_test, y\_pred, pos\_label=model.classes\_[1])

pr\_display = PrecisionRecallDisplay(precision=prec, recall=recall).plot(label=model)



from sklearn.metrics import confusion\_matrix

cf\_matrix = confusion\_matrix(Y\_test, X\_test\_prediction)

print(cf\_matrix)

[[41 4]

[ 4 65]]

tn, fp, fn, tp = cf\_matrix.ravel()

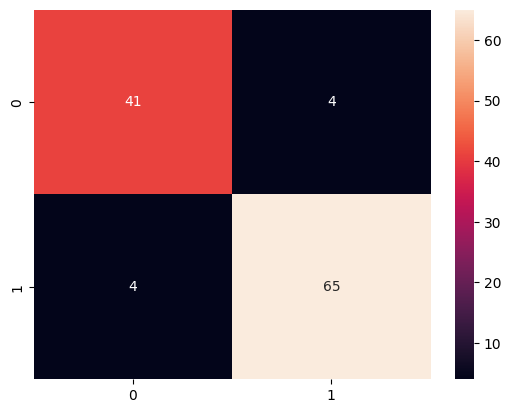
print(tn, fp, fn, tp)

41 4 4 65

import seaborn as sns

sns.heatmap(cf\_matrix, annot=True)

<Axes: >



 cv\_score = cross\_val\_score(model, X\_train, Y\_train, cv=3)

print(cv\_score)

[0.92763158 0.95394737 0.95364238]