

BREAST CANCER PREDICTION

Melvin Oswald Sahaya Anbarasu

Nii Otu Tackie-Otoo

Annamalai Muthupalaniappan

Atharva Vichare



Introduction

- Breast cancer is the most common cancer amongst women, and some men in the world.
- It accounts for 25% of all cancer cases, and affected over 2.1 Million people last year alone.
- It starts when cells in the breast begin to grow out of control. These cells usually form tumors that can be seen via X-ray or felt as lumps in the breast area.
- The key challenges against its detection is how to classify tumors into malignant (cancerous) or benign(non cancerous) being a medical problem.



Problem Motivation

1. Predicting Breast Cancer using the various Machine Learning Models.
2. Observing how hyperparameters influence the model performance.



Related Works

In paper 1, the authors attempted to predict breast cancer on the same dataset as ours. They used kNN, Random Forest and Naive Bayes models whereas we have used Logistic Regression, kNN, SVM and Random Forest for the same task. In their Setup, instead of Naive Bayes, we have tried a Logistic Regression and Support vector machine. We were able to achieve similar results on the common models between our setups.

In Paper 2, The authors compares the performance of many machine learning models on different datasets related to breast cancer particularly on the dataset used in our project. Some of the models trained and evaluated include SVM, Adaboost, Naive Bayes, ANN. we were able to achieve similar results for the common models in our setup.





Dataset

- Data from [kaggle](#)
- Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass.









Dataset

- The Data consists of 30 predictor variables and one target variable - Diagnosis.

- a) radius (mean of distances from center to points on the perimeter)
- b) texture (standard deviation of gray-scale values)
- c) perimeter
- d) area
- e) smoothness (local variation in radius lengths)
- f) compactness ($\text{perimeter}^2 / \text{area} - 1.0$)
- g). concavity (severity of concave portions of the contour)
- h). concave points (number of concave portions of the contour)
- i). symmetry
- j). fractal dimension ("coastline approximation" - 1)



Dataset

	id int64 8670 - 911320502 	diagnosis object B 62.7% M 37.3%	radius_mean float64 6.981 - 28.11 	texture_mean float64 9.71 - 39.28 	perimeter_mean float64 43.79 - 188.5 	area_mean float64 143.5 - 2501.0 	smoothness_mean float64 0.05263 - 0.1634 
0	842302	M	17.99	10.38	122.8	1001	0.1184
1	842517	M	20.57	17.77	132.9	1326	0.08474
2	84300903	M	19.69	21.25	130	1203	0.1096
3	84348301	M	11.42	20.38	77.58	386.1	0.1425
4	84358402	M	20.29	14.34	135.1	1297	0.1003
5	843786	M	12.45	15.7	82.57	477.1	0.1278
6	844359	M	18.25	19.98	119.6	1040	0.09463
7	84458202	M	13.71	20.83	90.2	577.9	0.1189
8	844981	M	13	21.82	87.5	519.8	0.1273
9	84501001	M	12.46	24.04	83.97	475.9	0.1186
10	845636	M	16.02	23.24	102.7	797.8	0.08206

569 rows x 32 columns

Deepnote

Data - Preprocessing

- Data Preprocessing
 - Balance Data
 - Standardize Data
- Label Encoding
 - Binary Classification Task
 - Diagnosis: Malignant (M) – 1, Benign (B) – 0
- Feature Selection
 - Choose relevant features to be used for model.



Model's Overview

- Logistic Regression
- KNN
- Support Vector Machines
- Random Forest



Logistic Regression

Accuracy without hyperparameters tuning = 0.974

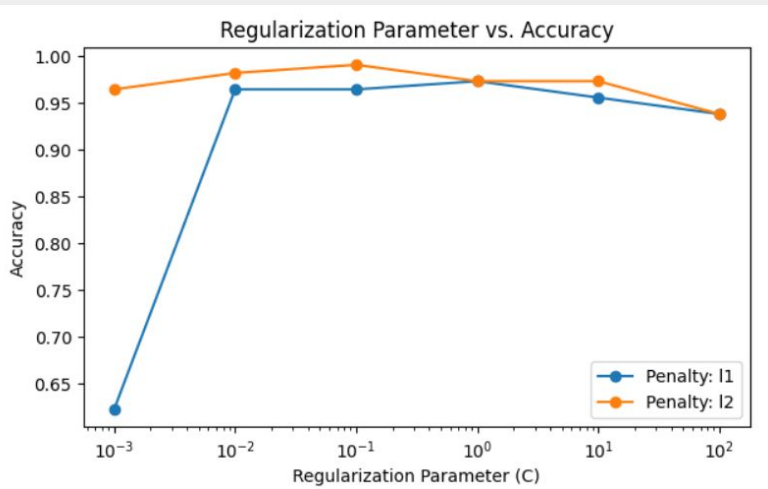
Hyperparameters

- Regularization Strength, $C = [0.001, 0.01, 0.1, 1, 10, 100]$
- Penalty = $[l1, l2]$

Best Hyperparameters:

- Regularization Strength $C = 0.1$
- Penalty = $l2$

Accuracy with best hyperparameters = 0.9912



KNN

Accuracy without hyperparameters tuning = 0.947

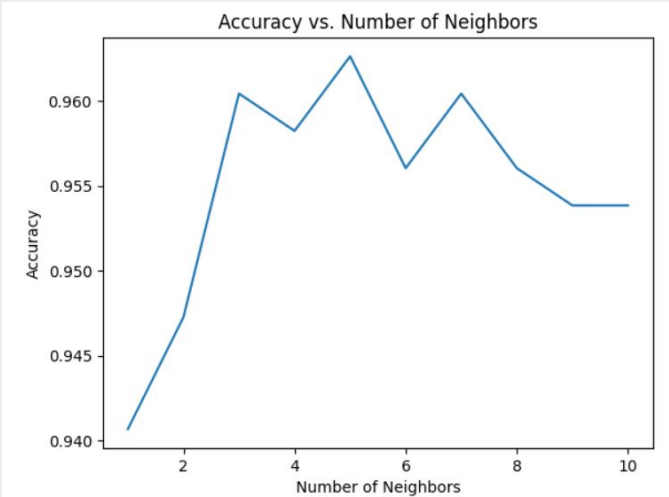
Hyperparameters

- Number of Neighbors
 - [2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
- Metric
 - Manhattan, Euclidean

Best Hyperparameters:

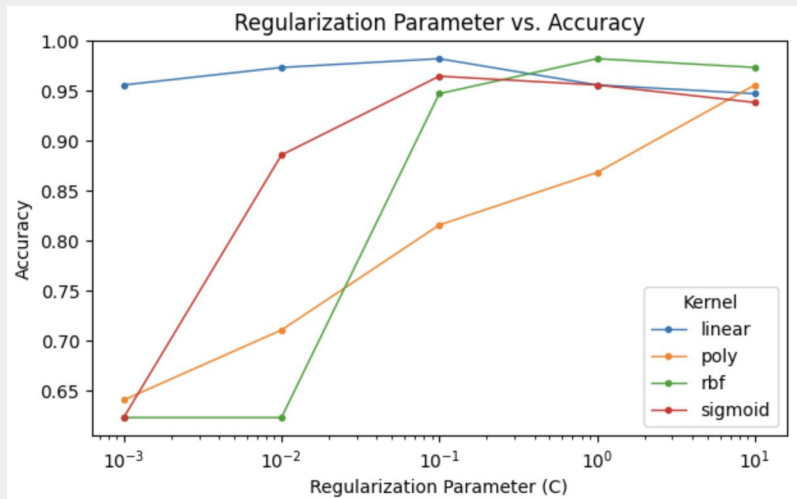
- Number of Neighbors = 5
- Metric = Euclidean

Accuracy with best hyperparameters = 0.965



Support Vector Machine

Accuracy without hyperparameters tuning = 0.9825



Hyperparameters

- Kernel
 - Linear, polynomial, radial basis function (rbf), sigmoid
- Regularization Parameter, C
 - 0.001, 0.01, 0.1, 1, 10

Best Hyperparameters:

- Kernel = linear
- Regularization Parameter, C = 0.1

Accuracy with best hyperparameters = 0.9825



Random Forest

Accuracy without hyperparameters tuning = 0.956

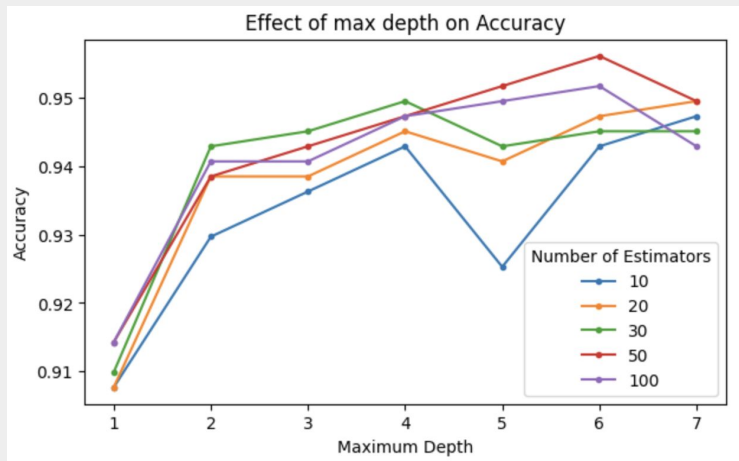
Hyperparameters

- Number of Estimators
 - [10, 20, 30, 50, 100]
- Max Depth
 - [1, 2, 3, 4, 5, 6, 7]

Best Hyperparameters:

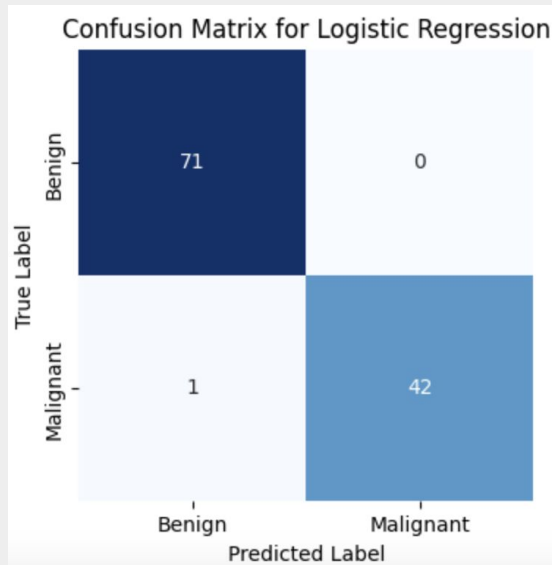
- Number of Estimators = 50
- Max Depth = 6

Accuracy with best hyperparameters = 0.965

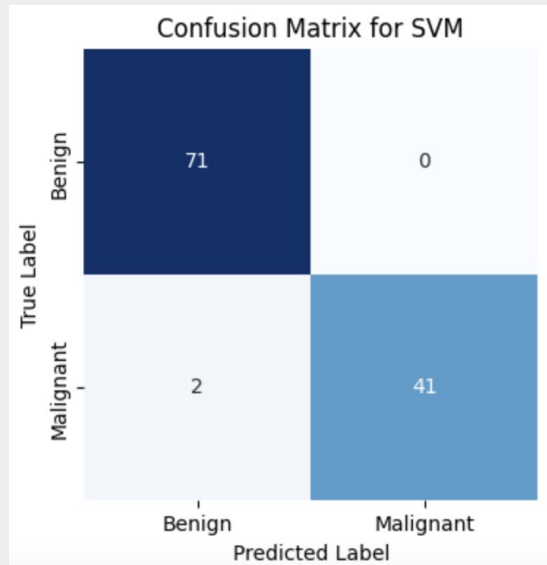


Confusion Matrix

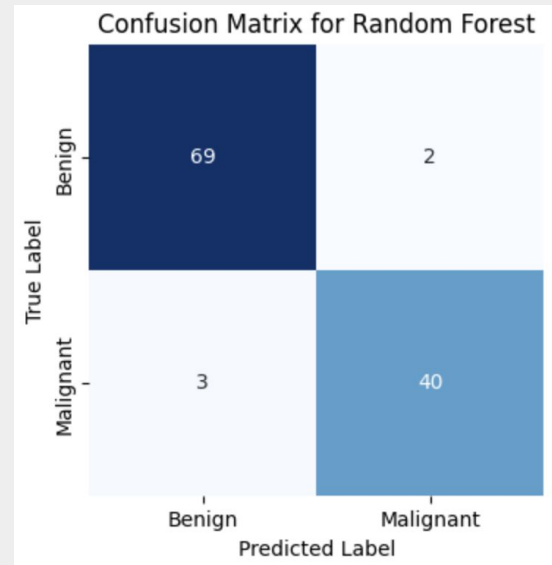
Considering confusion matrix of top 3 performing models



Accuracy = 0.9912
Recall = 0.98



Accuracy = 0.983
Recall = 0.95



Accuracy = 0.965
Recall = 0.93





Conclusion

1. For our dataset, Logistic Regression Model performed the best with the highest Recall on top of being the one with the highest accuracy.
2. By Playing around with hyperparameters, we came to know how they affect the model's performance.





Thank you all for Listening !

