**Project team**

|  |  |  |  |
| --- | --- | --- | --- |
| Team leader | 20P6785 | Matthew Sherif | 20p6785@eng.asu.edu.eg |
| Team member | 20P6794 | Karim Bassel | 20p6794@eng.asu.edu.eg |
| Team member | 20p7341 | Fady Fady Fouad | 20p7341@eng.asu.edu.eg |
| Team member | 20p7239 | Shady Emad Sabry | 20p7239@eng.asu.edu.eg |

**Introduction and Background**

The project aims to develop a machine learning model that can accurately diagnose breast cancer using various clinical and demographic features. Machine learning algorithms can identify patterns and predict disease outcomes with high accuracy, making them a promising tool for improving breast cancer diagnosis  
  
**Objectives**

* Develop a machine learning model that can accurately diagnose breast cancer using various clinical and demographic features.
* Evaluate the performance of the model using standard metrics such as accuracy, precision, recall, and F1-score.
* Compare the performance of multiple machine learning algorithms to identify the most effective algorithm for breast cancer diagnosis.

**System overview**

* System takes Dataset as an input.
* Dataset is split to training and testing data
* Training data is trained among several models then tested
* Accuracy of each model is calculated
* Confusion matrix for each model is shown to see how well the model performed on each case(True positive, True negative, False positive, False negative)

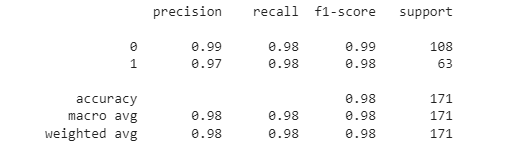
**Methods**

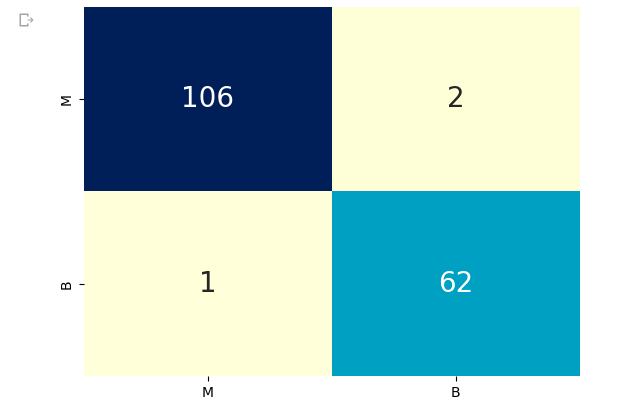
4 training models have been used

* Logistic Regression
  + It works by fitting a logistic function to the input data, which maps the input features to a probability of belonging to a certain class. The logistic function is a sigmoidal curve that ranges from 0 to 1, with 0.5 being the decision boundary. The model learns the optimal values of the coefficients for the logistic function using optimization techniques such as gradient descent. Once trained, the model can predict the probability of a new input belonging to a certain class based on the learned coefficients and decision boundary.
* DecisionTreeClassifier
  + It works by recursively splitting the input data based on the values of input features to create a tree-like structure that makes decisions. At each level of the tree, the algorithm chooses the feature that provides the most information gain or reduction in impurity to split the data. The splitting process continues until a stopping criterion is met, such as a maximum tree depth or minimum number of samples per leaf. Once trained, the model can predict the target variable of a new input by traversing the tree structure based on the values of its input features.
* RandomForestClassifier
  + It works by creating multiple decision trees using a random subset of the input features and training them on a random subset of the input data. The trees are grown using the same process as the Decision Tree algorithm. Once trained, the model can predict the target variable of a new input by aggregating the predictions of the individual trees. The aggregation can be done through majority voting or averaging, depending on the type of problem being solved. Random Forest Classifier is known for its ability to handle high-dimensional data and reduce overfitting, making it a popular choice for classification tasks.
* KNN
  + It works by finding the K closest data points in the training set to a new input and using their target variable values to predict the target variable of the new input. The distance between the input and training data points is typically calculated using Euclidean distance. Once the K closest neighbors are identified, the model can predict the target variable of the new input by taking the majority vote or averaging the target variable values of the K nearest neighbors. KNN is known for its simplicity and ability to handle nonlinear data, making it a popular choice for classification and regression tasks.

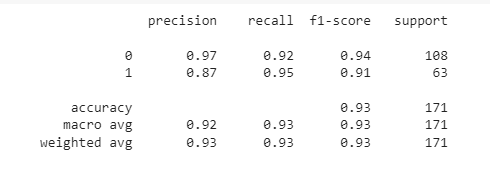
**Results and Evaluation**

At the end of each model accuracy, Classification report, Confusion Matrix are all output. And here are the output samples

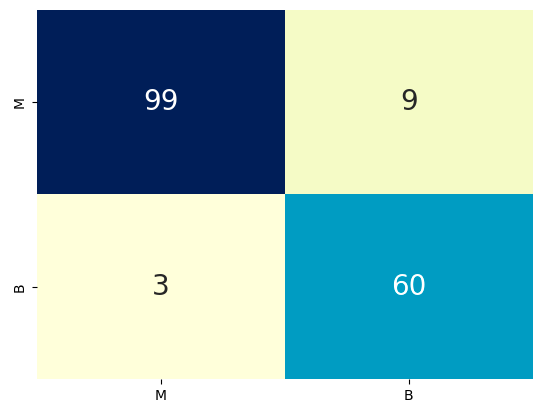
* Logistic Regression
  + Accuracy: 0.9824561403508771
  + CR:
  + Confusion Matrix:



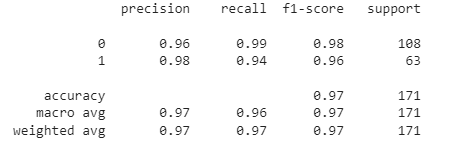
* DecisionTreeClassifier
  + Accuracy 0.9298245614035088
  + CR:



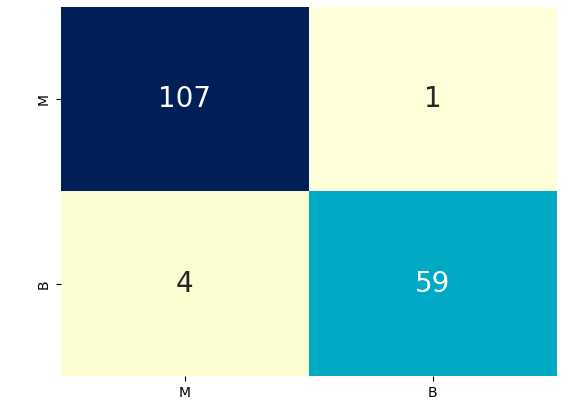
* + Confusion Matrix:



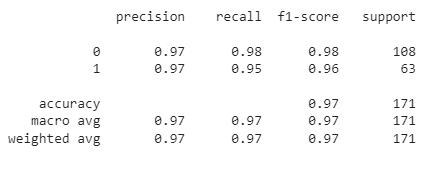
* RandomForestClassifier
  + Accuracy: 0.9707602339181286
  + CR:



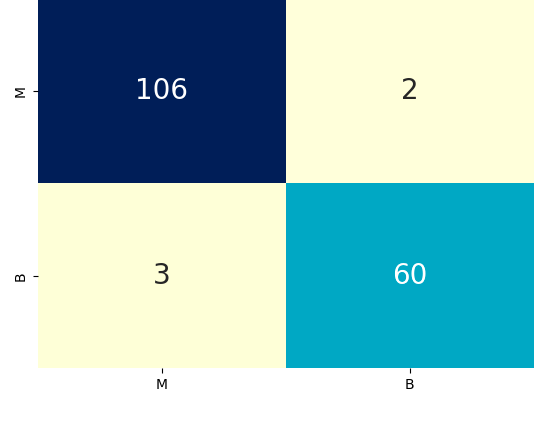
* + Confusion Matrix:



* KNN
  + Accuracy: 0.9707602339181286
  + CR:

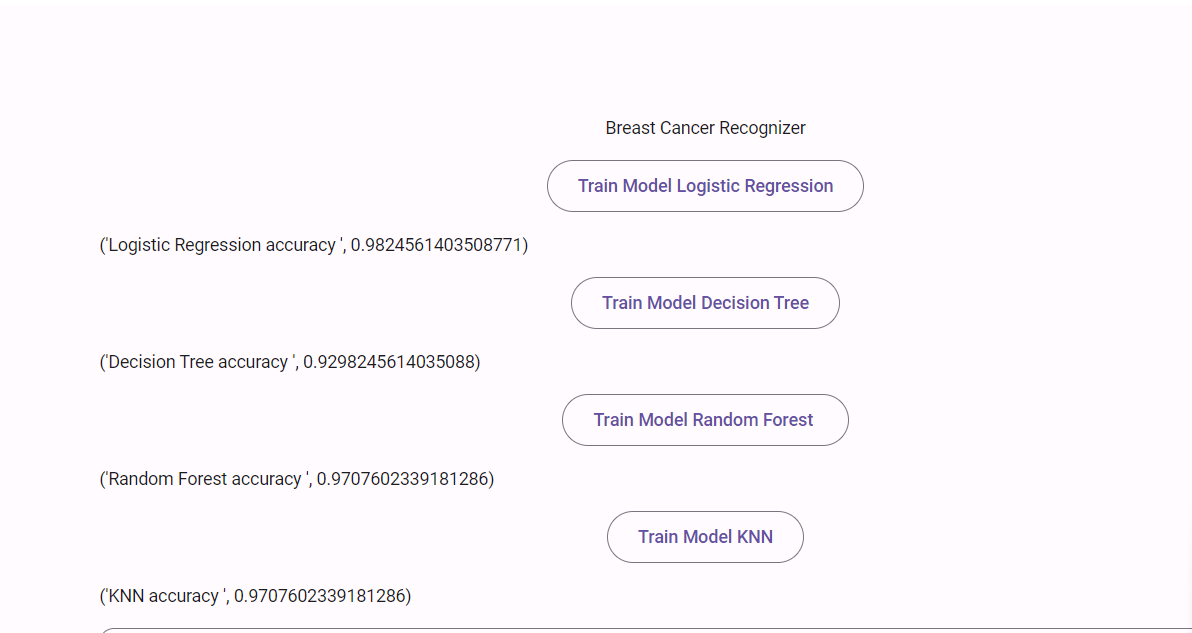


* + Confusion Matrix:

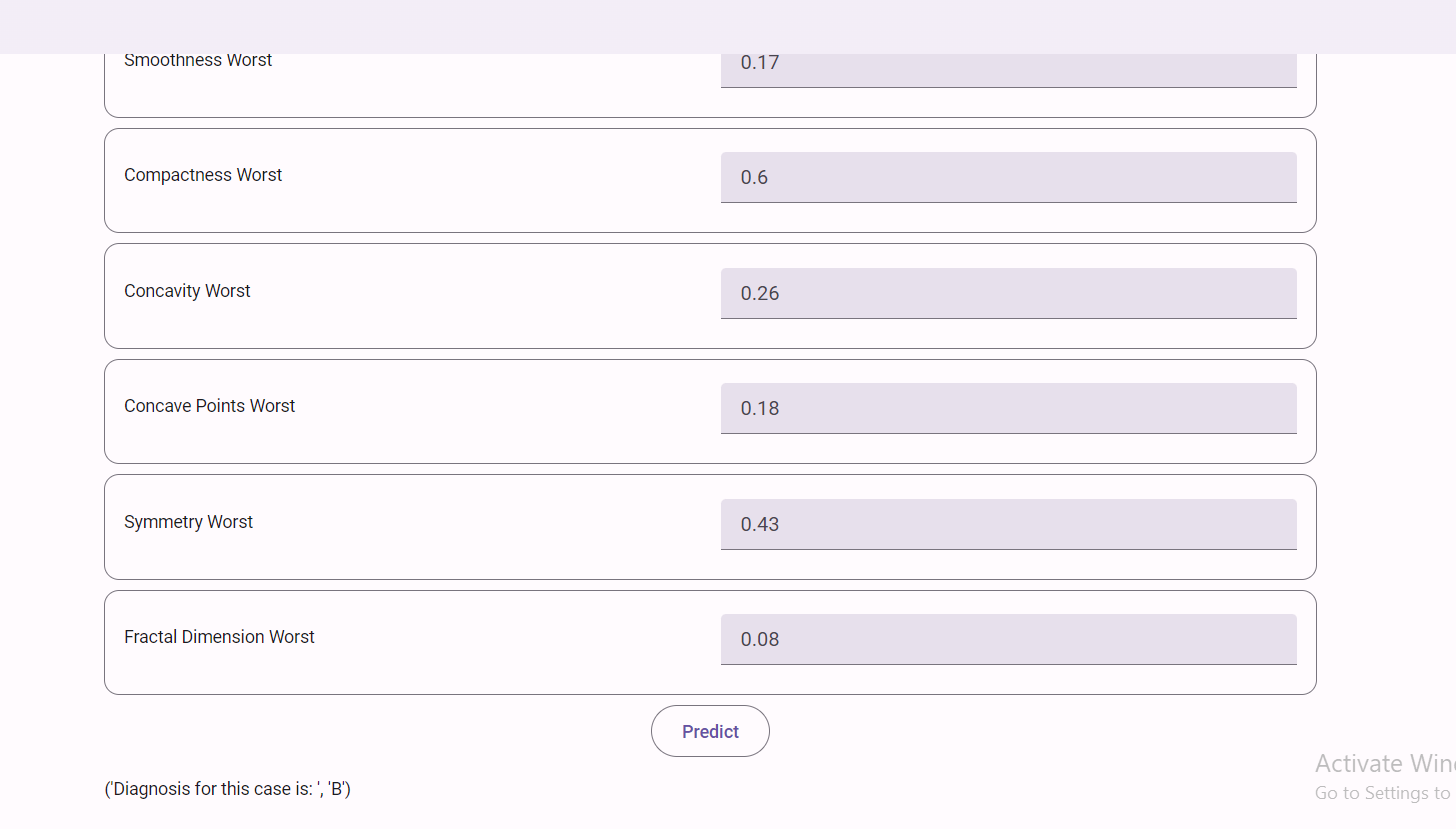


**GUI**

**Anvil web application used**

****

The 4 models accuracies

****

Diagnosis predicted for specific input

**Grading Criteria**

You should follow the minimum requirements of the project and the grading is based on the discussion as a group as well as each team member in the group. The grading distribution can be follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Point** | **Out of** | **Grade** | **Notes** |
| Does the report follow the required outline? | **3** |  |  |
| Application/System is running & all required modules are implemented | **3** |  |  |
| Is the dataset presented as a benchmark or collected? | **3** |  |  |
| Does the model/system clearly specify the loss/metric used to evaluate the quality of results? | **3** |  |  |
| Is the final model used to solve the problem obtained by selecting one of at least 2 candidate models? | **3** |  |  |
| Quality of discussion/conclusions. | **5** |  |  |

**Approval Signatures**

|  |  |  |
| --- | --- | --- |
| Supervisor |  | Teaching Assistance |
| Associate Professor / Walaa H. Elashmawi |  | Eng. Ziad Elgayar |

**Grade:**

20