BRAC UNIVERSITY

Inspiring Excellence

CSE422: Artificial Intelligence

Report: Detection of Breast Cancer

Submitted by Group 4
Tasnia Hossain (23341097)
Raisa Tahasen (21201284)
Ishika Mehrab (21201792)
Mohammad Sazidur Rahman (21301473)

Table of Contents

Introduction

The predominant aim of this project is to develop a model that detects whether or not tumor masses are benign or malignant, and thereby accurately diagnose breast cancer, through the predictive analyses of relevant data. Four different models are put to use to achieve this end: Decision Tree, Random Forest, Logistic Regression, and the K-Nearest Neighbors algorithm.

Dataset Description

Source:

Original Link:
https://www.kaggle.com/datasets/yasserh/breast-cancer-dataset?fbclid=IwAR1GJpPhSiRIUVp6V5DVB-OUEDyY7kp2ZmZQktleDXSQ2uJHAg7jLOO9Gts

Modified Link:
https://drive.google.com/file/d/1dz9l_GNMUCh-M09TJX0jGlt0G-6eU6pe/view?usp=sharing

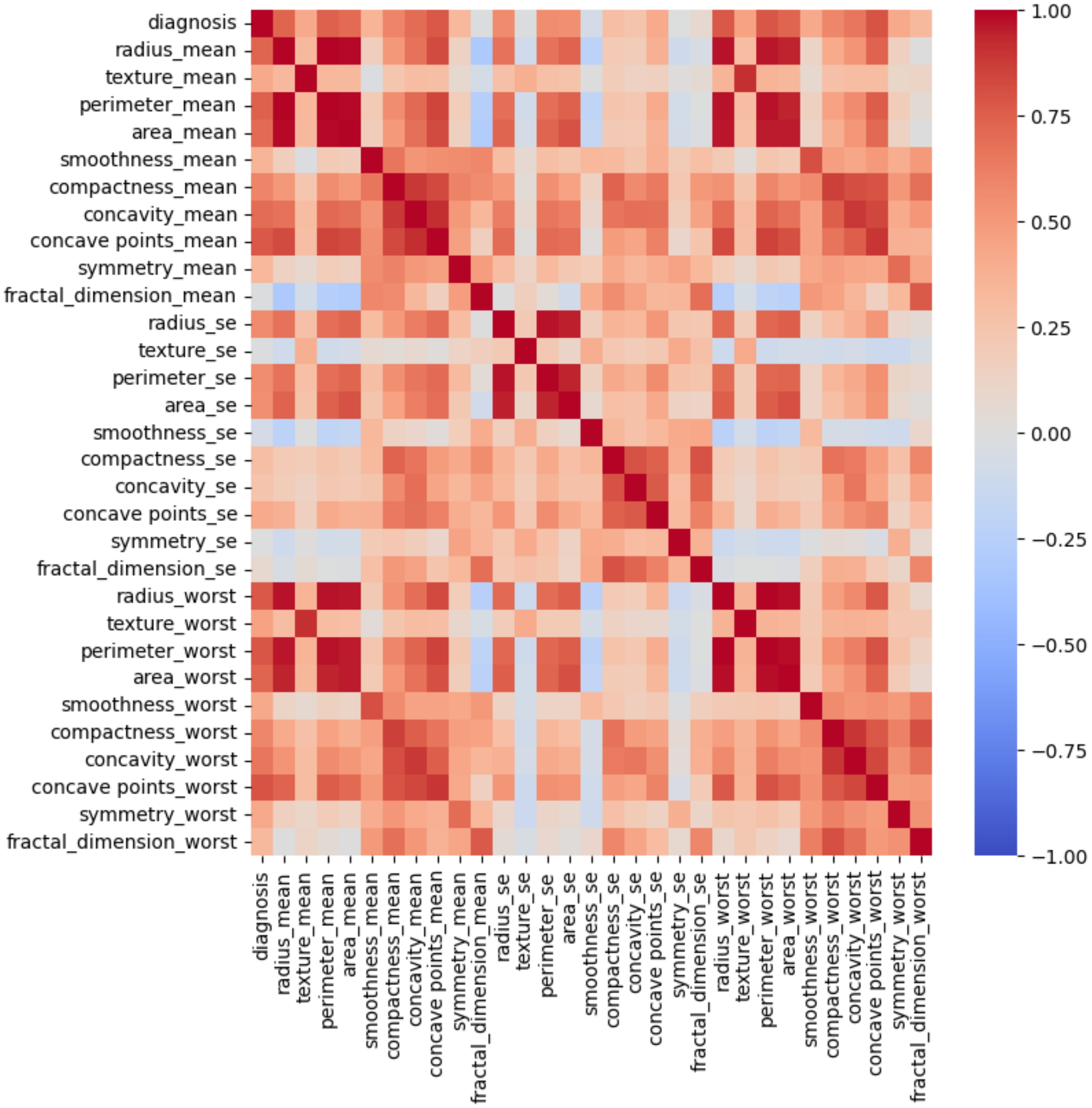
Reference: https://ieeexplore.ieee.org/abstract/document/8410258

- Number of features: The dataset has 32 features. The first 2 (“id” and “diagnosis”) are attribute information, and the rest 30 are the means, standard errors and worsts of the following:
    1. Radius
    2. Texture
    3. Perimeter
    4. Area
    5. Smoothness
    6. Compactness
    7. Concavity
    8. Concave points
    9. Symmetry
    10. Fractal dimension

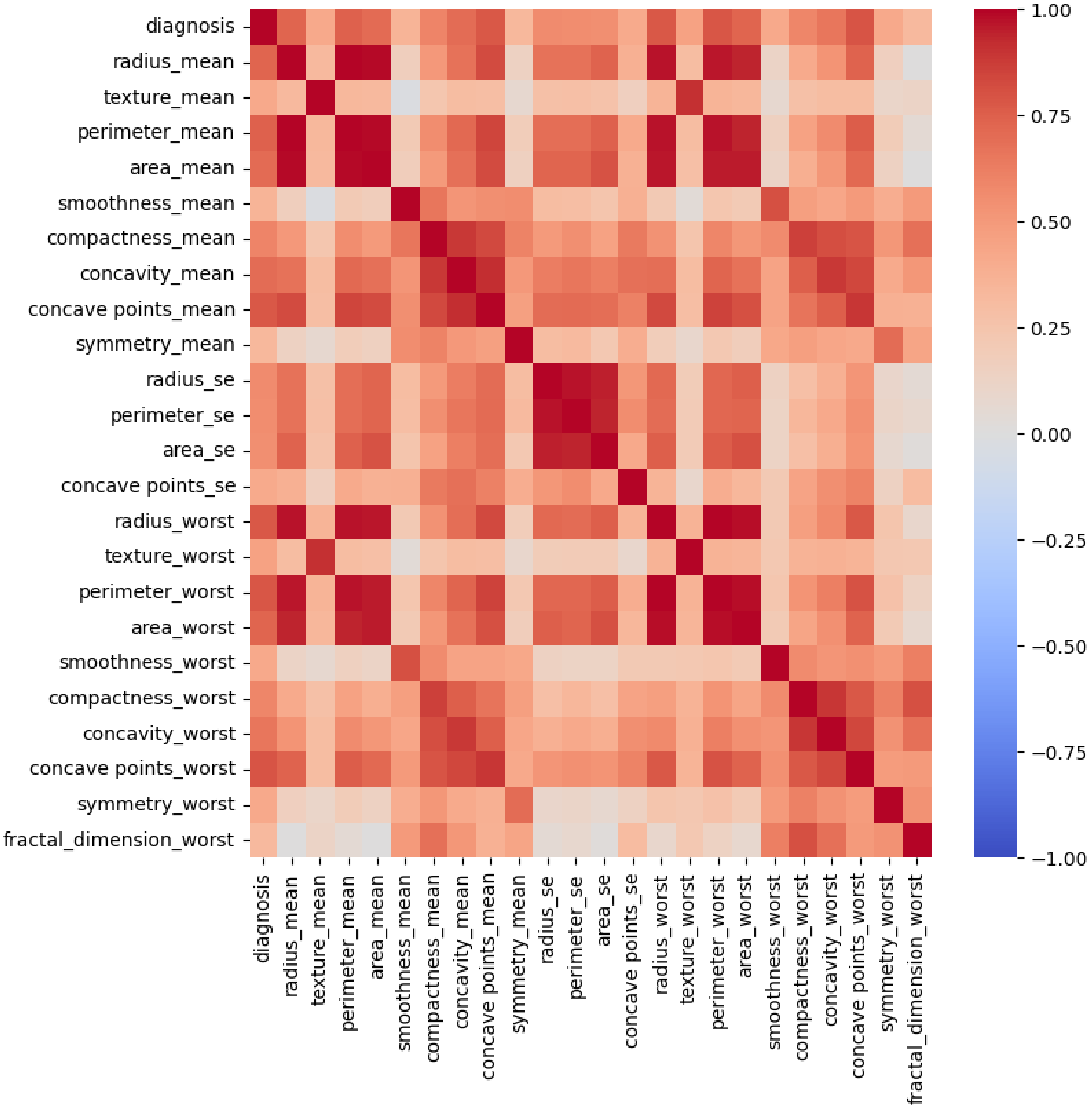
- Problem type: This is a classification problem because it seeks to identify whether irregular cellular masses are cancerous (malignant) or non-cancerous (benign). With the presence of cancer (M) being depicted as 1 in the diagnosis, and the absence of it (B) as 0, this can be categorized as a binary classification problem.
- Number of datapoints: 576
- Feature description: Most of the features of this dataset are quantitative and continuous. These features hold numerical data or real numbers, but the target variable “diagnosis” is a categorical feature because it holds binary values B and M.

- Feature correlation (input and output features): Before we begin, we check the correlation between our target variable “diagnosis” and other features. Since “id” cannot be used for classification, it doesn’t require analysis, so we drop it. We utilize the heatmap from the seaborn library to understand the degree of correlation between the features and select only the more strongly connected features for further testing.

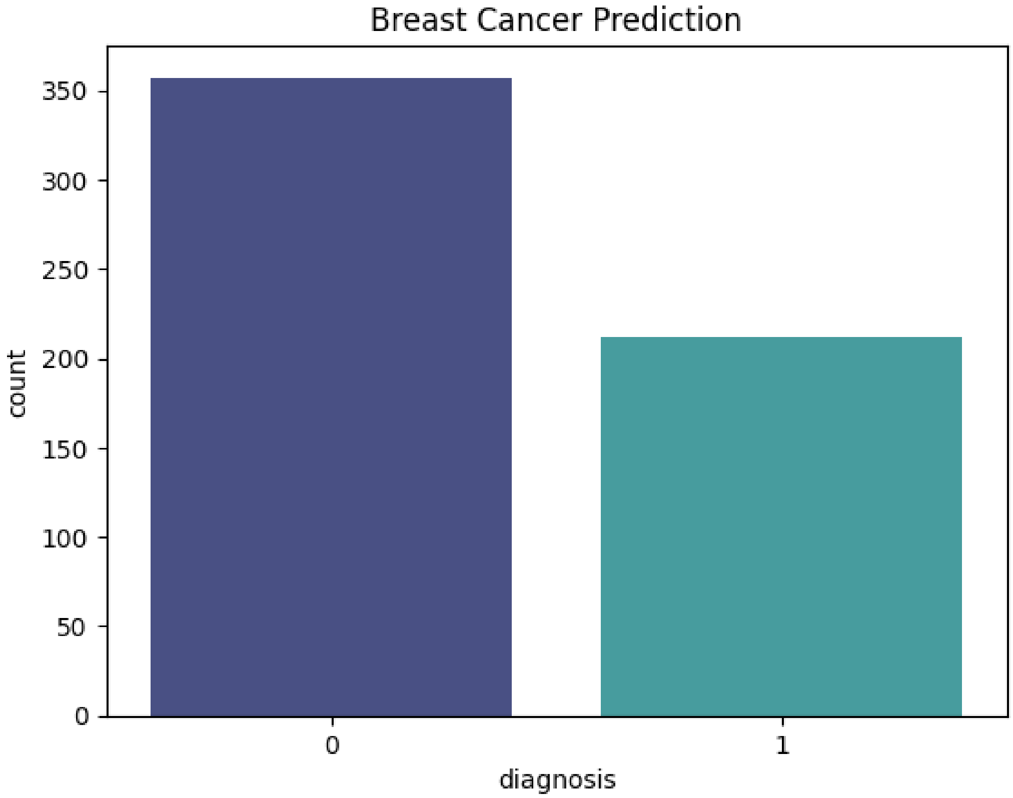
  The following diagram displays the heatmap before the dropping of weakly connected features/columns:



The following diagram displays the heatmap after the dropping of weakly connected features/columns, such as "fractal_dimension_mean", "fractal_dimension_se", "smoothness_se", "symmetry_se", etc.:



- Imbalanced Dataset: In the output feature of our dataset, the unique classes do not have an equal number of instances as represented in the bar chart below.



There are significantly more diagnoses of benign tumor masses (0) than there are of malignant ones (1). The dataset is thus imbalanced.

Dataset Pre-processing

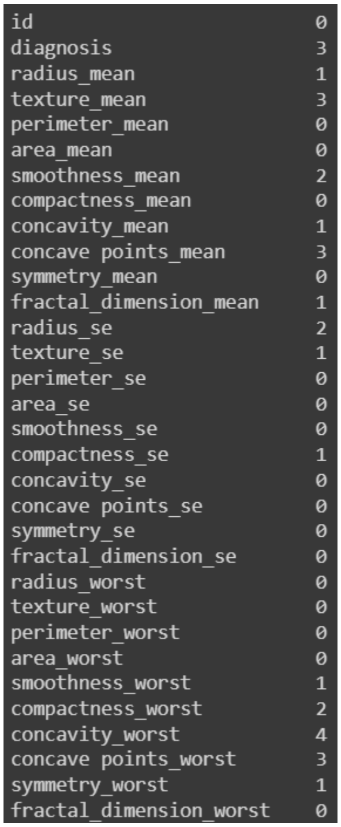
- **Dealing with NULL values**

  Using the pandas library to read through our dataset, we first identified the number of null values for each column in the original data.

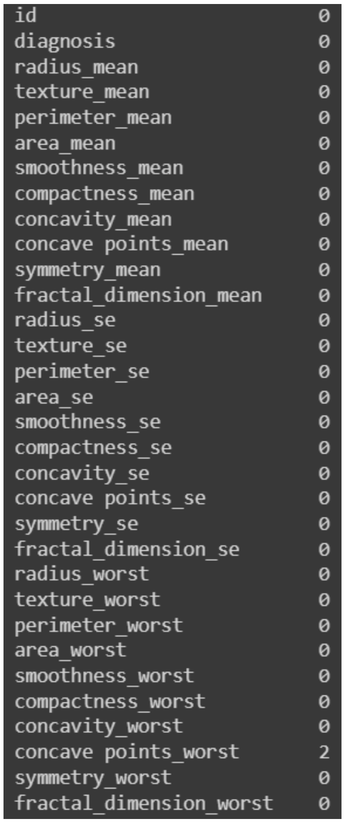
  Since the “diagnosis” column was our target variable, there could be no null values in any of its rows; we thus dropped all rows with null values in the target variable column.

  Apart from that, other columns contained very few null values, but we couldn't drop entire columns because of just this. We instead looked into the number of null values for each row and dropped any row containing more than 1 null value since data of a patient with more than one feature value is unreliable.

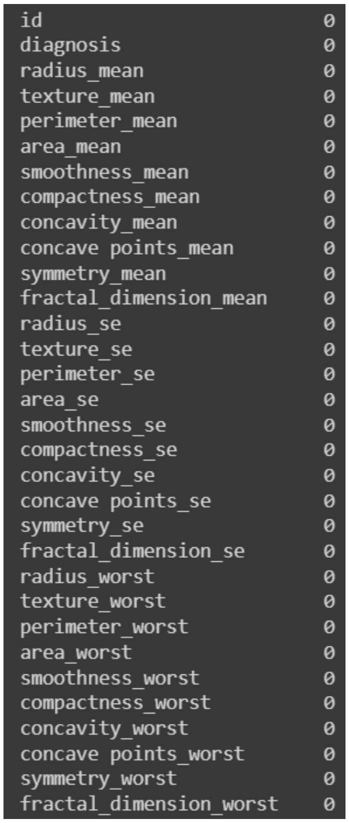
  Finally, we were only left with rows with only one null value and those missing values were imputed utilizing the sklearn library. We used median to do this since medians are unaffected by extreme values. These chronological changes have been shown below:



```
id                         0
diagnosis                  3
radius_mean                1
texture_mean               3
perimeter_mean             0
area_mean                  0
smoothness_mean            2
compactness_mean           0
concavity_mean             1
concave points_mean        3
symmetry_mean              0
fractal_dimension_mean     1
radius_se                  2
texture_se                 1
perimeter_se               0
area_se                    0
smoothness_se              0
compactness_se             1
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           1
compactness_worst          2
concavity_worst            4
concave points_worst       3
symmetry_worst             1
fractal_dimension_worst    0
```

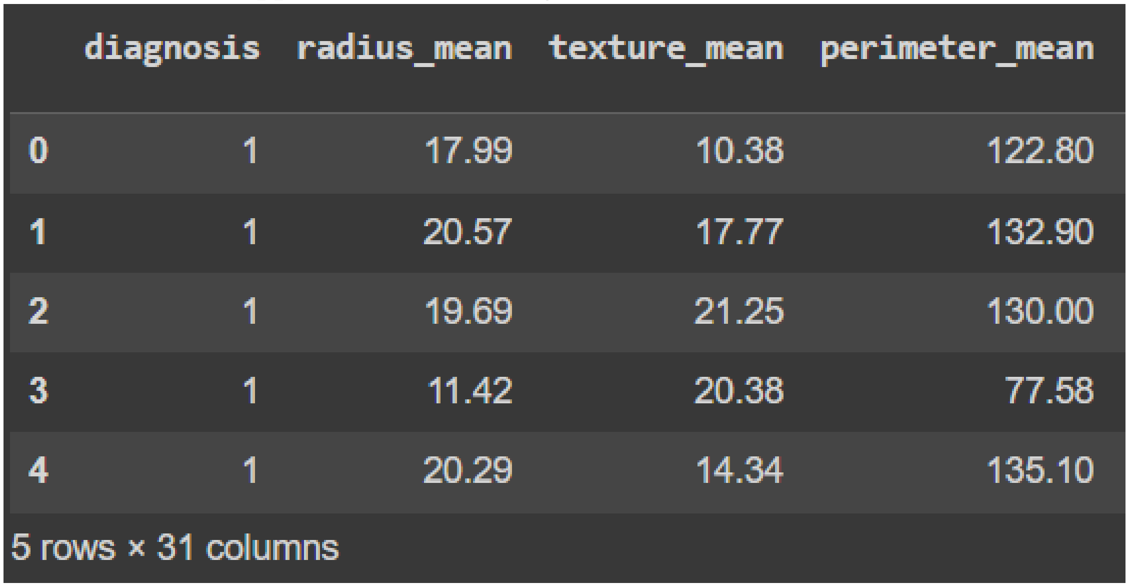


```
id                         0
diagnosis                  0
radius_mean                0
texture_mean               0
perimeter_mean             0
area_mean                  0
smoothness_mean            0
compactness_mean           0
concavity_mean             0
concave points_mean        0
symmetry_mean              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       2
symmetry_worst             0
fractal_dimension_worst    0
```



```
id                         0
diagnosis                  0
radius_mean                0
texture_mean               0
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
```

- **Encoding**
  The target variable “diagnosis” is categorical data, containing binary value B or M and thus it was mapped as the following: B=0, M=1.



	diagnosis	radius_mean	texture_mean	perimeter_mean
0	1	17.99	10.38	122.80
1	1	20.57	17.77	132.90
2	1	19.69	21.25	130.00
3	1	11.42	20.38	77.58
4	1	20.29	14.34	135.10

5 rows × 31 columns

Feature Scaling

Although feature scaling wasn’t required for our other models, we used the StandardScaler() function in the KNN model that we employed.

Dataset Splitting:

We split the dataset into 70% training dataset and 30% testing dataset because this ratio gave us the best results. We set the random state to 100 so that the same split is used every time, as this made it convenient for us when comparing the models.

Model Training & Testing:

The following four models were used to train and test our data:

- **Decision Tree:** Decision trees are tree-like structures where each node represents a decision based on a feature, leading to subsequent nodes until a prediction is made. Decision trees can be used to identify key features (such as tumor size, age, etc.) and their thresholds to classify whether a breast cancer case is malignant or benign.

- **Random Forest:** Random Forest is an ensemble learning method that builds multiple decision trees and combines their predictions to improve accuracy and reduce overfitting. Random Forest can enhance the robustness of predictions by aggregating the results of 1000 decision trees in our model, thereby improving the accuracy of breast cancer classification.

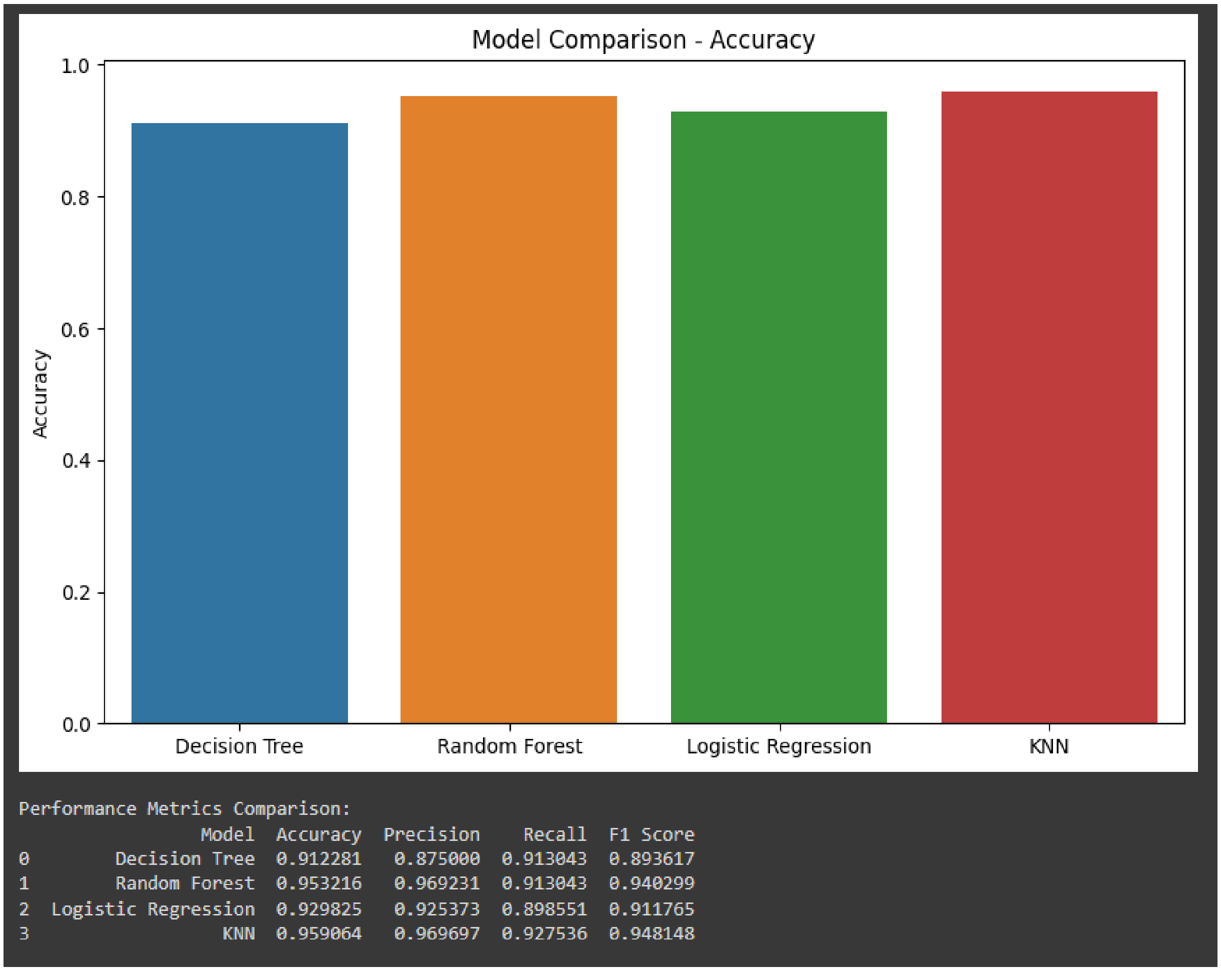
- **Logistic Regression:** Logistic regression models the probability of a binary outcome using the logistic function. It is a type of linear model suitable for binary classification problems. Logistic regression can be applied to predict the likelihood of breast cancer being malignant based on input features. It provides probabilities and a threshold is set to make the final classification decision.

- **KNN:** KNN classifies data points based on the majority class of their k-nearest neighbors in the feature space. It's a non-parametric and instance-based learning algorithm. In our breast cancer prediction project, KNN would classify a new case by examining the class labels of its 3-nearest neighbors, making predictions based on the majority class. Since one feature had values higher than others, it dominated the classification determination. So we scaled our data first for better results.

In the first three cases, data did not require scaling, so in the training phase, the models were fitted on the training data we obtained after processing and splitting the original data. In the case of KNN, as mentioned earlier, the data was scaled, and then the model was fitted onto it. The trained models were then evaluated using the separate test sets in each of the cases.

Model selection/Comparison analysis:

- **Bar Chart:**
  The bar chart below illustrates the comparison of accuracy between Decision Tree, Random Forest, Logistic Regression, and KNN. All the models have been trained and tested accordingly with the dataset, and their performances are evaluated based on their accuracy in predicting breast cancer.



```
Performance Metrics Comparison:
                 Model  Accuracy  Precision    Recall  F1 Score
0        Decision Tree  0.912281   0.875000  0.913043  0.893617
1        Random Forest  0.953216   0.969231  0.913043  0.940299
2  Logistic Regression  0.929825   0.925373  0.898551  0.911765
3                  KNN  0.959064   0.969697  0.927536  0.948148
```

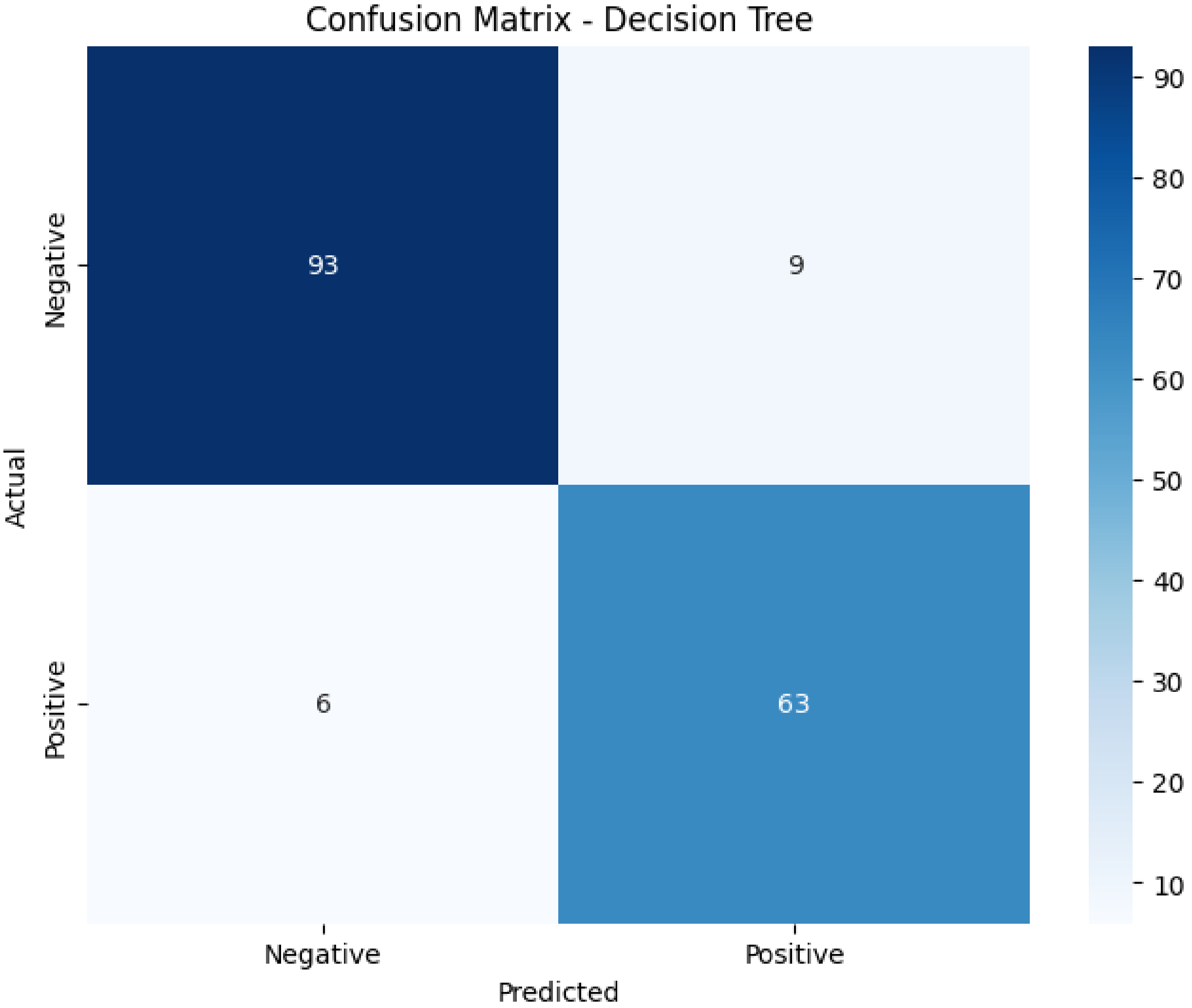
According to the accuracy comparison, the KNN model exceeds all other models with an accuracy of 95.9%. Furthermore, by comparing other measures such as precision, recall, and F1 score, we can gain further insight into the models' capacity to detect breast cancer. Despite the fact that Random Forest and KNN are almost similar and have the highest precision among the other models, the overall F1 score implies that the KNN model is the best of all of them.

- **Confusion Matrix:**

1. **For Decision Tree model:**

   There were 93 True negatives and 63 True positives in this case. This is the number of patients who were appropriately classified, i.e. breast cancer detection was done correctly.

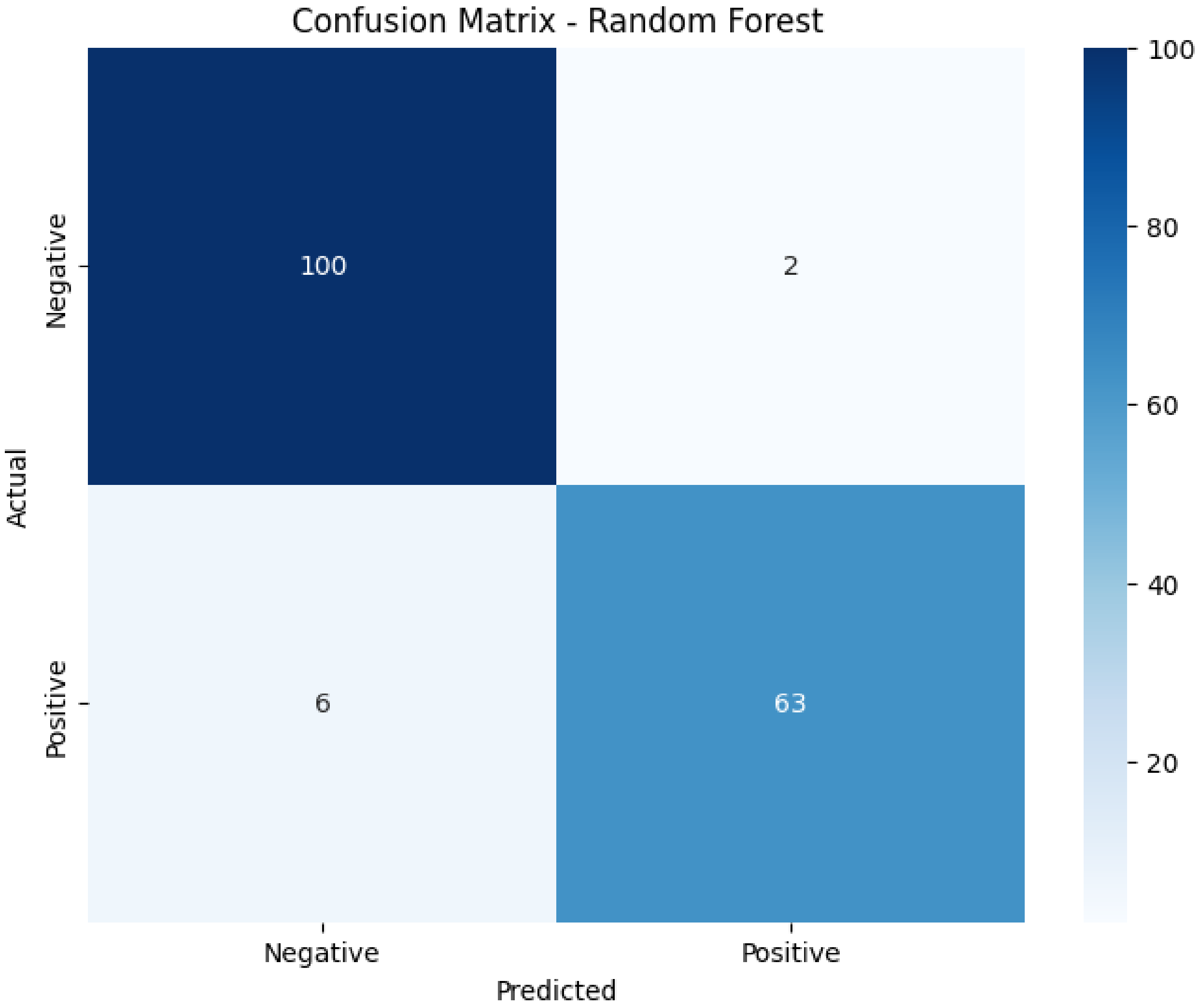
   Six patients were misdiagnosed as not having cancer when they actually did. Similarly, 9 patients were misclassified as having cancer when, in fact, they did not.



2. **For Random Forest model:**

There were 100 True negatives and 63 True positives in this case. These are the number of patients who were appropriately classified, i.e. breast cancer detection was done correctly.

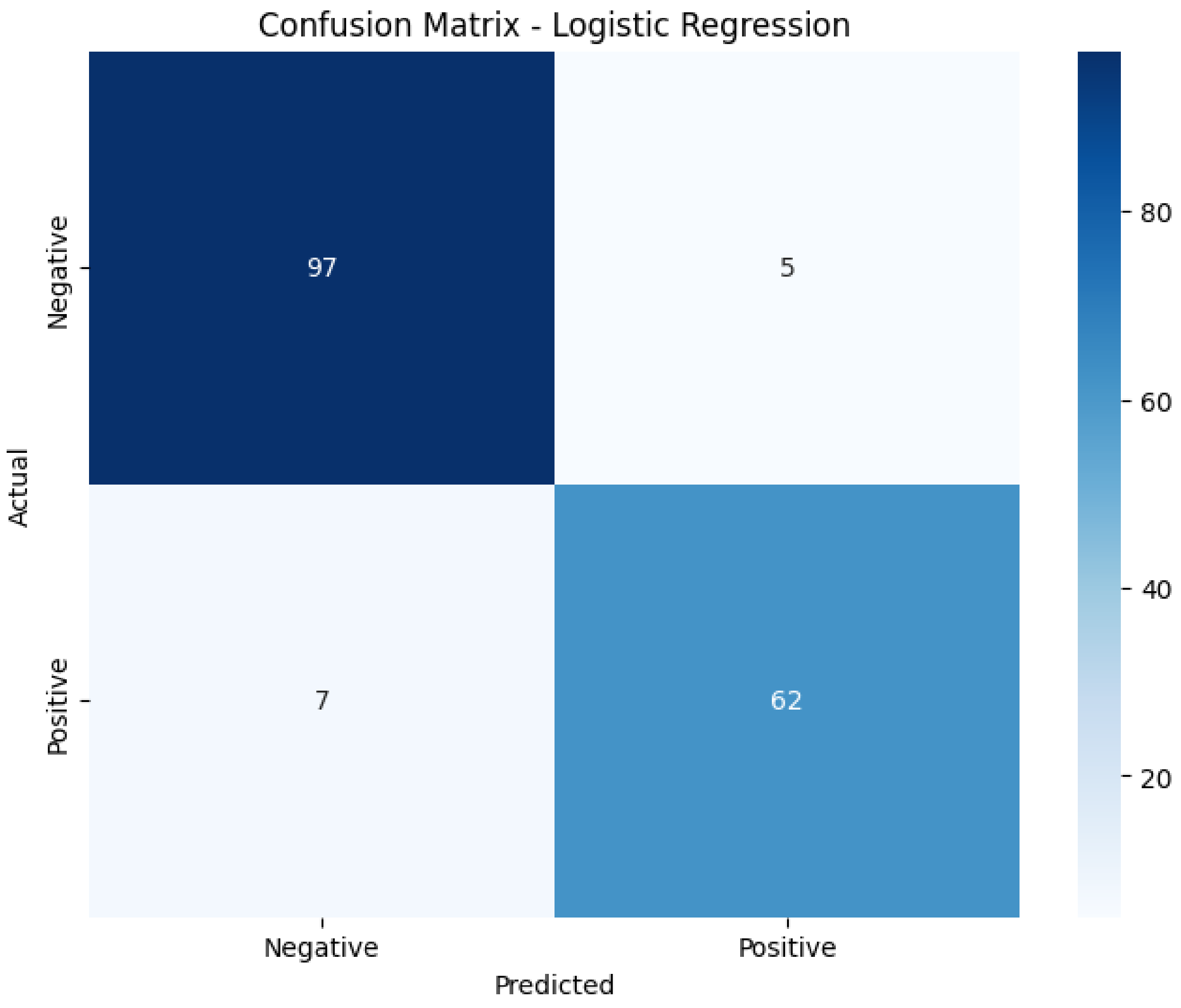
Six patients were misdiagnosed as not having cancer but they actually did. Similarly, 2 patients were misclassified as having cancer when, in fact, they did not.



3. **<u>For Logistic Regression model:</u>**

There were 97 True negatives and 62 True positives in this case. These are the number of patients who were appropriately classified, i.e. breast cancer detection was done correctly.

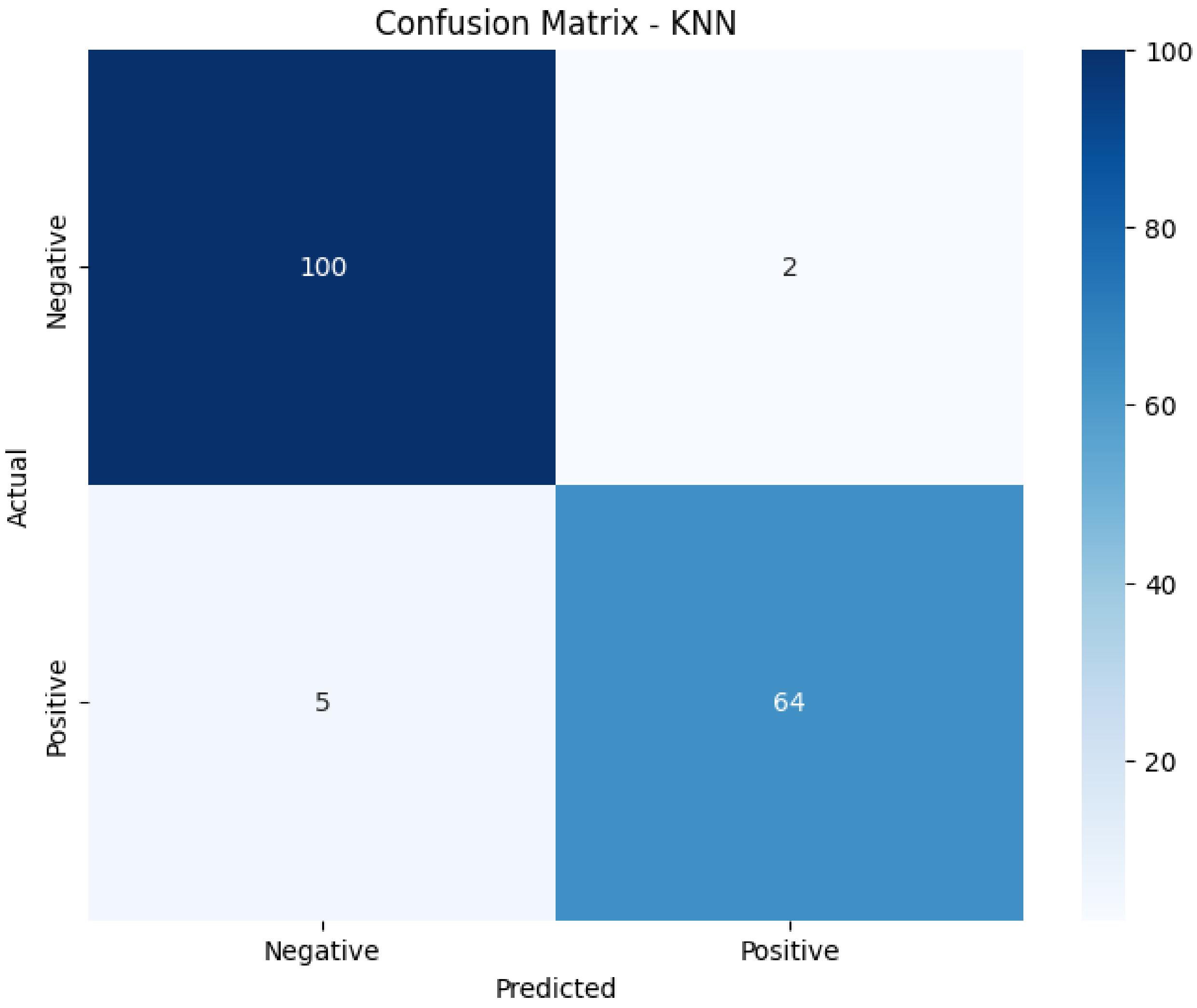
7 patients were misdiagnosed as not having cancer but they actually did. Similarly, 5 patients were misclassified as having cancer when, in fact, they did not.



4. **<u>For KNN model:</u>**

   There were 100 True negatives and 64 True positives in this case. These are the number of patients who were appropriately classified, i.e. breast cancer detection was done correctly.

   5 patients were misdiagnosed as not having cancer but they actually did. Similarly, 2 patients were misclassified as having cancer when, in fact, they did not.



Conclusion:

As per the analysis of our data, the KNN model produces the most accurate predictions when diagnosing breast cancer, with a 95.9% accuracy score, 96.9% precision score, 92.5% recall and 94.8% F1 score. We can conclude that our project on the detection of breast cancer has proven to yield quite strong and reliable results. The advancement of this kind of technology will undoubtedly revolutionize the scope of medicine, the speed of diagnoses, and the quality of treatment that patients receive.