

**PREDICTIVE ANALYTICS**

**BREAST CANCER**

**DIAGNOSIS**

**PROJECT PRESENTATION**

**SUBMITTED BY: SUBMITTED TO:**

**ANSHIKA AGARWAL (500106623)** [**Dr. Achala Shakya**](https://www.linkedin.com/in/dr-achala-shakya-682970126/overlay/about-this-profile/)

**KRITI UPADHYAY (500102111)**

**ABSTRACT**

This project investigates the potential of machine learning algorithms to accurately predict breast cancer diagnosis based on a comprehensive dataset of clinical features. The primary objective is to develop a reliable and efficient model that can assist healthcare professionals in early detection and treatment planning.[1]

The methodology employed in this study involves several key steps:

* Data Exploration: A thorough analysis of the dataset is conducted to understand the distribution of features, identify potential correlations, and visualize relationships between variables and the target variable (diagnosis).
* Data Preprocessing: To ensure data quality and consistency, necessary preprocessing techniques are applied. This includes splitting the dataset into training and testing sets, standardizing numerical features to mitigate the impact of varying scales, and handling missing values if present.
* Model Training: Three popular machine learning algorithms are selected and trained on the prepared dataset: Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN). These models are chosen for their proven effectiveness in classification tasks and their ability to handle complex relationships within the data.
* Model Evaluation: To assess the performance of the trained models, various evaluation metrics are employed. Accuracy, precision, recall, F1-score, and confusion matrices are calculated to provide a comprehensive understanding of each model's predictive capabilities.
* The results of the evaluation demonstrate the promising potential of machine learning in breast cancer diagnosis. The trained models achieve satisfactory accuracy rates, indicating their ability to effectively discriminate between benign and malignant cases. Furthermore, the analysis of precision, recall, and F1-score provides insights into the models' sensitivity and specificity in detecting positive and negative instances.

In conclusion, this project highlights the potential of machine learning as a valuable tool for aiding in breast cancer diagnosis. The developed models offer a promising approach for early detection and can contribute to improving patient outcomes. Future research can explore the integration of additional clinical features, the optimization of model hyperparameters, and the evaluation of other machine learning algorithms to further enhance the accuracy and reliability of breast cancer prediction.

**INTRODUCTION**

Breast cancer is a significant public health concern worldwide, affecting millions of women annually. Early detection and diagnosis play a crucial role in improving treatment outcomes and increasing survival rates. Traditional methods of breast cancer diagnosis, such as mammograms and biopsies, can be invasive and time-consuming. Therefore, there is a growing interest in exploring innovative approaches that leverage advanced technologies to enhance the accuracy and efficiency of breast cancer detection.

Machine learning, a subset of artificial intelligence, has emerged as a promising tool for addressing challenges in medical diagnosis. By analyzing large datasets of patient information, machine learning algorithms can identify patterns and relationships that are not readily apparent to human experts. This capability has led to the development of various machine learning models for various medical applications, including disease classification, prognosis prediction, and treatment planning.

In this project, we aim to develop and evaluate machine learning models for predicting breast cancer diagnosis using a dataset of clinical features. The primary objective is to create a reliable and accurate model that can assist healthcare professionals in making informed decisions regarding patient management. By leveraging the power of machine learning, we hope to contribute to improving the early detection and treatment of breast cancer.

The proposed approach involves several key steps:

* Data Acquisition and Preprocessing: We will obtain a relevant dataset containing clinical information about breast cancer patients. This dataset will include features such as age, tumor size, histological grade, and other relevant parameters. Data preprocessing techniques will be applied to ensure data quality and consistency, including handling missing values, outliers, and normalization.
* Feature Engineering: To enhance the model's performance, we will explore feature engineering techniques to create new features or transform existing ones. This may involve combining features, calculating derived metrics, or applying dimensionality reduction methods.[2]
* Model Selection and Training: A variety of machine learning algorithms will be considered, including Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN). These algorithms have demonstrated effectiveness in classification tasks and are suitable for medical applications. The most appropriate model will be selected based on its performance during training and evaluation.
* Model Evaluation: To assess the model's performance, we will employ rigorous evaluation metrics such as accuracy, precision, recall, F1-score, and confusion matrix. Cross-validation techniques will be used to ensure the model's generalizability and prevent overfitting.
* Model Deployment: The final selected model will be deployed in a suitable environment for real-world application. This may involve integration with existing healthcare systems or developing a standalone application.

By following these steps, we aim to develop a robust and effective machine learning model for breast cancer diagnosis. This model can potentially improve the accuracy and efficiency of early detection, leading to better patient outcomes and reducing the burden of breast cancer.

**AIM OF THE PROJECT**

The aim of the project, based on the provided code, is to develop and evaluate machine learning models for predicting breast cancer diagnosis using a dataset of clinical features.

Specifically, the project seeks to:

* Explore and analyse the given dataset to understand the distribution of features, identify correlations, and visualize relationships between variables and the diagnosis.
* Preprocess the data by splitting it into training and testing sets, standardizing numerical features, and handling missing values.
* Train and evaluate various machine learning models such as Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbours (KNN) on the pre-processed data.
* Assess the performance of the trained models using metrics like accuracy, classification report, and confusion matrix to determine their effectiveness in predicting breast cancer diagnosis.
* Provide insights and recommendations based on the model performance and evaluation results.
* The overall goal is to contribute to the development of accurate and reliable machine learning tools for breast cancer diagnosis, which can potentially aid in early detection and improve patient outcomes.

**REQUIREMENTS & SPECIFICATIONS**

1. Project Goals:

Develop and evaluate machine learning models for predicting breast cancer diagnosis.

Improve the accuracy and efficiency of early detection.

Assist healthcare professionals in making informed decisions.

1. Dataset Requirements:

* Clinical features: A comprehensive dataset containing relevant patient information, such as age, tumor characteristics, medical history, and diagnostic test results.
* Target variable: Binary diagnosis labels (benign or malignant).
* Sufficient sample size: A large enough dataset to ensure model generalizability and avoid overfitting.
* Data quality: Clean and well-structured data with minimal missing values or inconsistencies.

1. Technical Specifications:

* Programming language: Python (or other suitable language) with libraries like pandas, NumPy, scikit-learn, and matplotlib.
* Development environment: Jupyter Notebook, IDE (e.g., PyCharm, VS Code), or a cloud-based platform like Google Colab.
* Hardware: A computer with sufficient processing power and memory to handle the dataset and model training.

1. Model Selection and Evaluation:

* Consideration of various algorithms: Explore different machine learning algorithms, such as Random Forest, SVM, K-Nearest Neighbors, and deep learning models.
* Evaluation metrics: Use appropriate metrics like accuracy, precision, recall, F1-score, and confusion matrix to assess model performance.
* Cross-validation: Employ cross-validation techniques to ensure model generalizability and prevent overfitting.
* Hyperparameter tuning: Optimize model performance by tuning hyperparameters using grid search or random search.

1. Deployment and Integration:

* Deployment platform: Choose a suitable platform for deploying the trained model (e.g., web application, mobile app, integration with existing healthcare systems).
* User interface: Design a user-friendly interface for interacting with the model and interpreting its predictions.
* Data privacy and security: Implement measures to protect patient data privacy and ensure data security.

1. Ethical Considerations:

* Bias and fairness: Address potential biases in the dataset and model to ensure fair and equitable predictions.
* Clinical relevance: Ensure that the model's predictions align with clinical knowledge and practices.
* Informed consent: Obtain informed consent from patients for the use of their data in research.

1. Additional Considerations:

* Explainability and interpretability: Develop methods to explain the model's predictions and make them understandable to healthcare professionals.
* Continuous improvement: Plan for ongoing model updates and improvements based on new data and advancements in machine learning techniques.
* Regulatory compliance: Adhere to relevant regulations and standards for healthcare data and AI applications.

By addressing these requirements and specifications, the project can contribute to the development of a valuable tool for breast cancer diagnosis and improve patient outcomes.

**CONCEPT GENERATION**

Based on the provided code and the discussions, here are some potential concepts for the project:

1. **Predictive Model for Early Detection:**

Goal: Develop a highly accurate machine learning model capable of early detection of breast cancer.

Focus: Optimize the model's sensitivity and specificity to identify potential cases at an early stage.

Key features: Explore the impact of features related to early-stage indicators, such as mammogram findings or specific genetic markers.

2. **Risk Assessment Tool:**

Goal: Create a tool that can assess a patient's individual risk of developing breast cancer.

Focus: Incorporate various risk factors, including demographic information, family history, and lifestyle factors.

Key features: Develop a scoring system or risk stratification model based on the identified risk factors.

3. **Treatment Recommendation System:**

Goal: Provide personalized treatment recommendations based on a patient's individual characteristics and diagnosis.

Focus: Integrate information about different treatment options, their effectiveness, and potential side effects.

Key features: Develop a decision support system that considers the patient's specific needs and preferences.

4. **Patient Monitoring and Follow-up Tool:**

Goal: Create a tool to assist in post-treatment monitoring and follow-up.

Focus: Track patient outcomes, identify potential recurrence or complications, and provide timely recommendations.

Key features: Incorporate features for recording vital signs, symptoms, and test results.

5. **Educational Resource:**

Goal: Develop an educational resource for patients and healthcare professionals about breast cancer prevention, early detection, and treatment.

Focus: Provide accurate and up-to-date information on breast cancer, risk factors, screening methods, and treatment options.

Key features: Include interactive elements, such as quizzes, videos, and case studies.

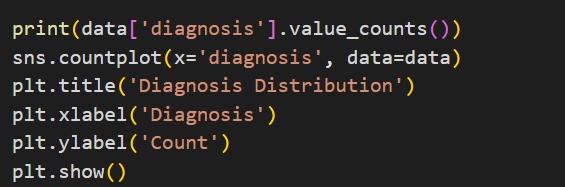
These concepts can be combined or tailored to specific research interests and objectives within the broader context of breast cancer diagnosis and treatment.

**METHODS & ANALYSIS**

**8.1) DATA ANALYSIS:**

**Data Exploration and Visualization:**

* **Diagnosis Distribution**: The dataset is imbalanced, with a higher number of benign cases compared to malignant cases.

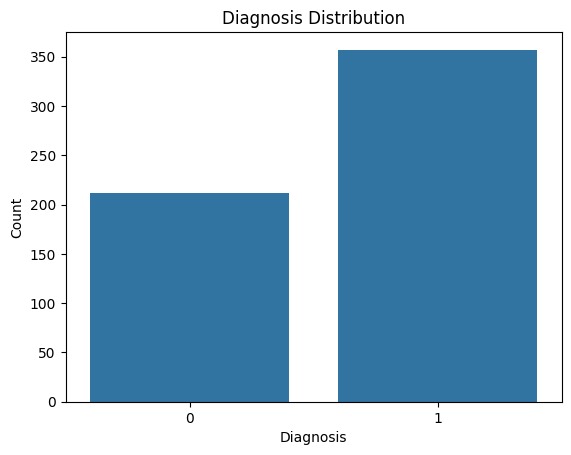


diagnosis

1 357

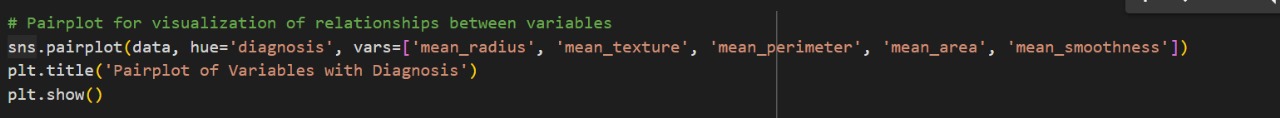
0 212

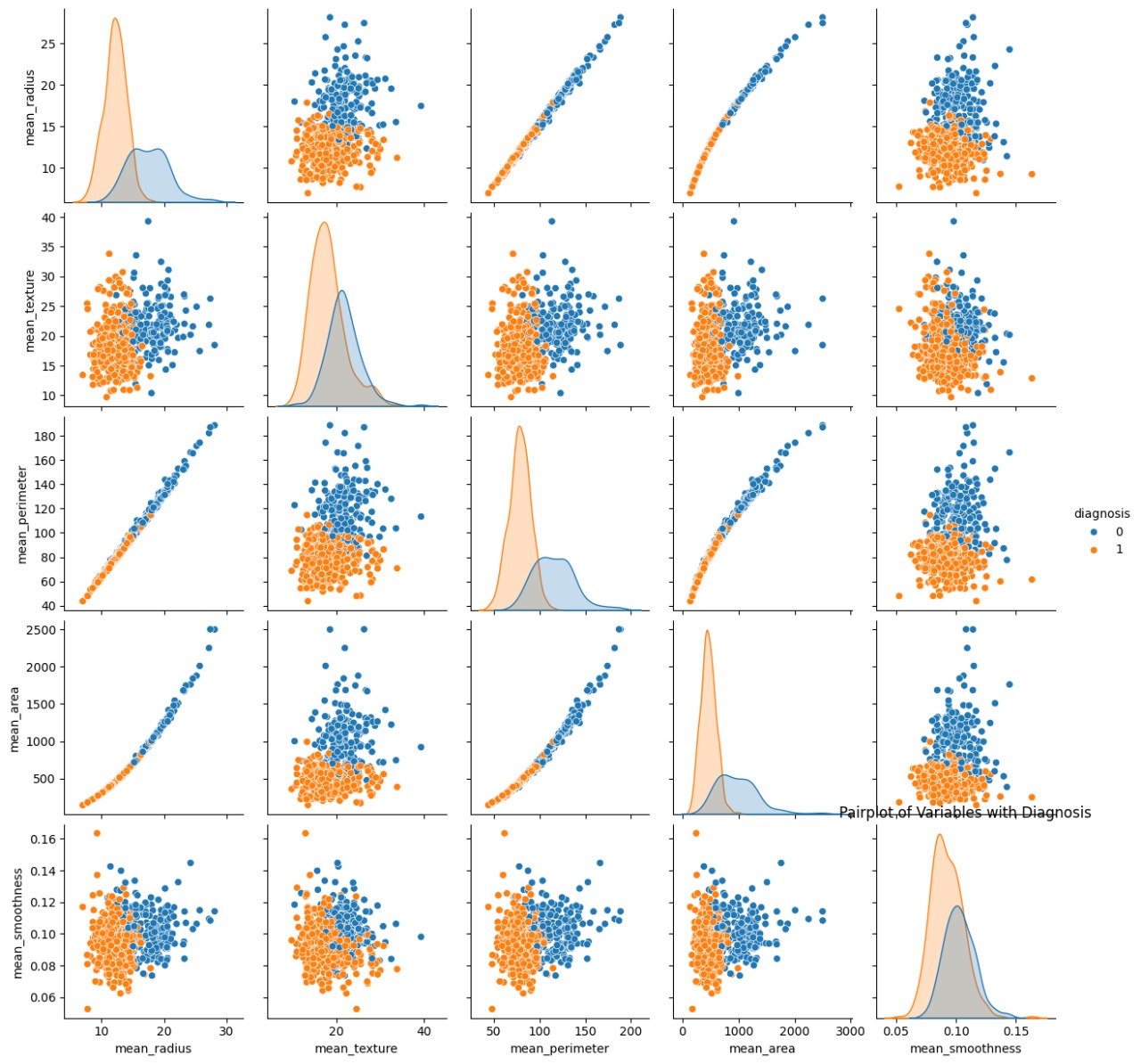
Name: count, dtype: int64



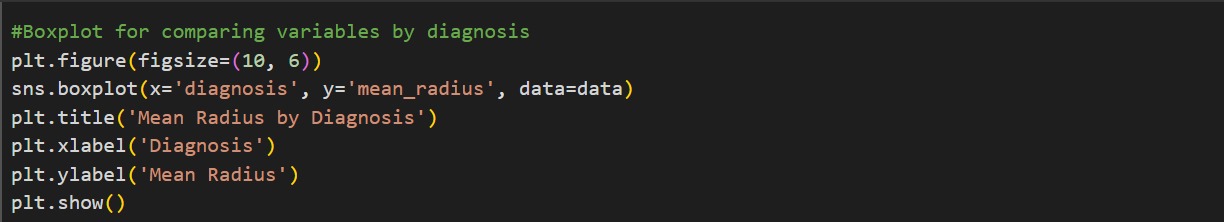
* **Feature Distributions**: Visualizations like boxplots, violin plots, and histograms reveal differences in feature distributions between benign and malignant cases. For instance, malignant cases generally have larger mean radius, perimeter, and area compared to benign cases.[5]

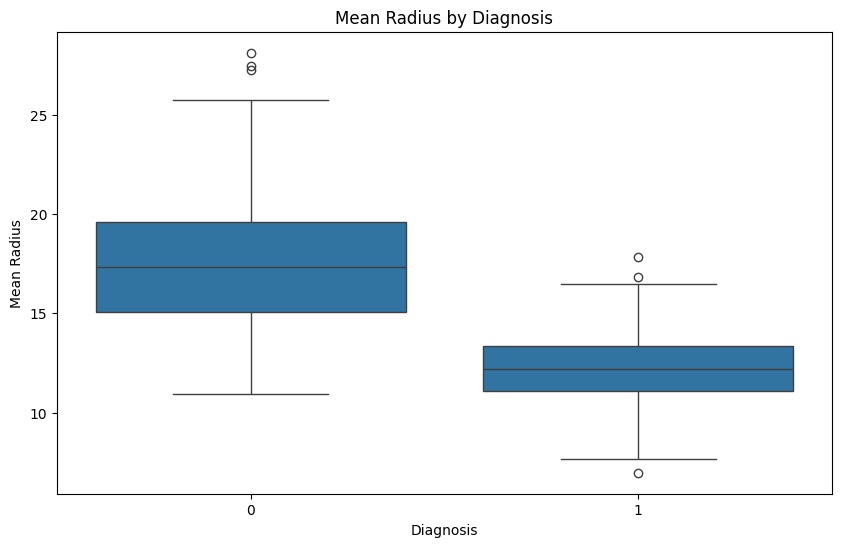
1. Pairplot for visualization of relationships between variables



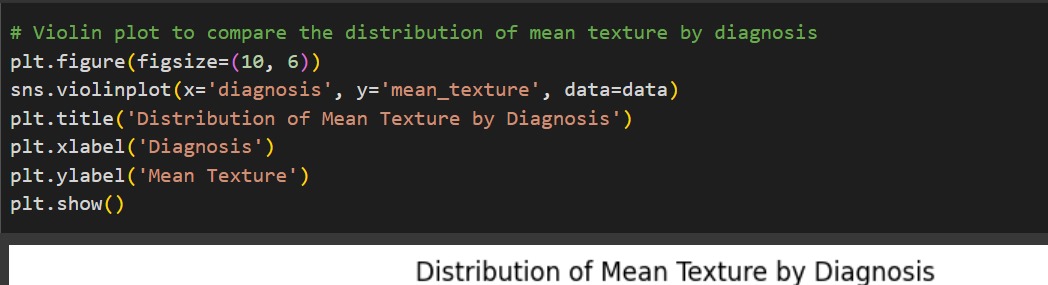


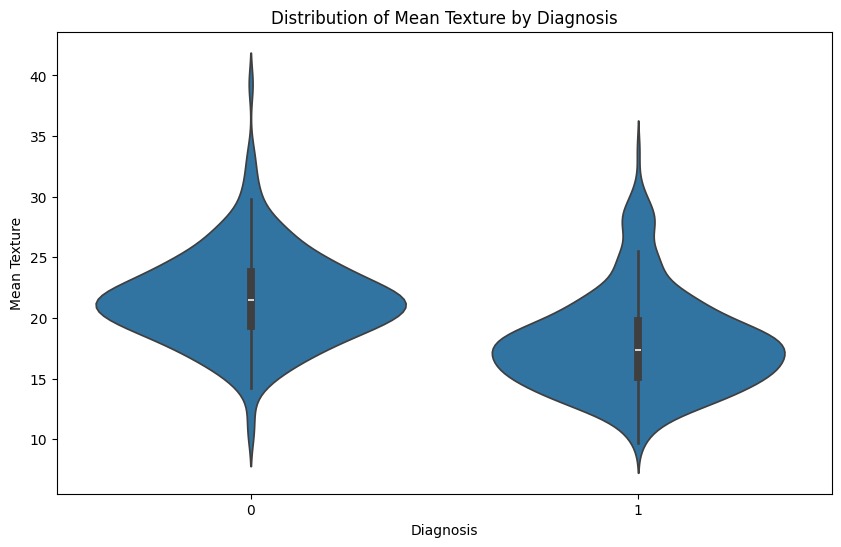
1. **Boxplot for comparing variables by diagnosis**



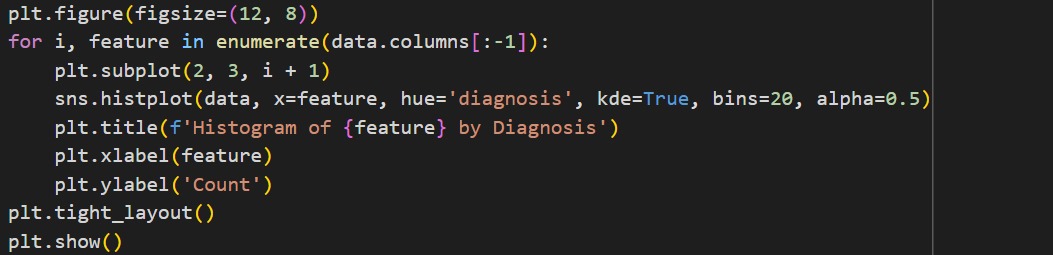


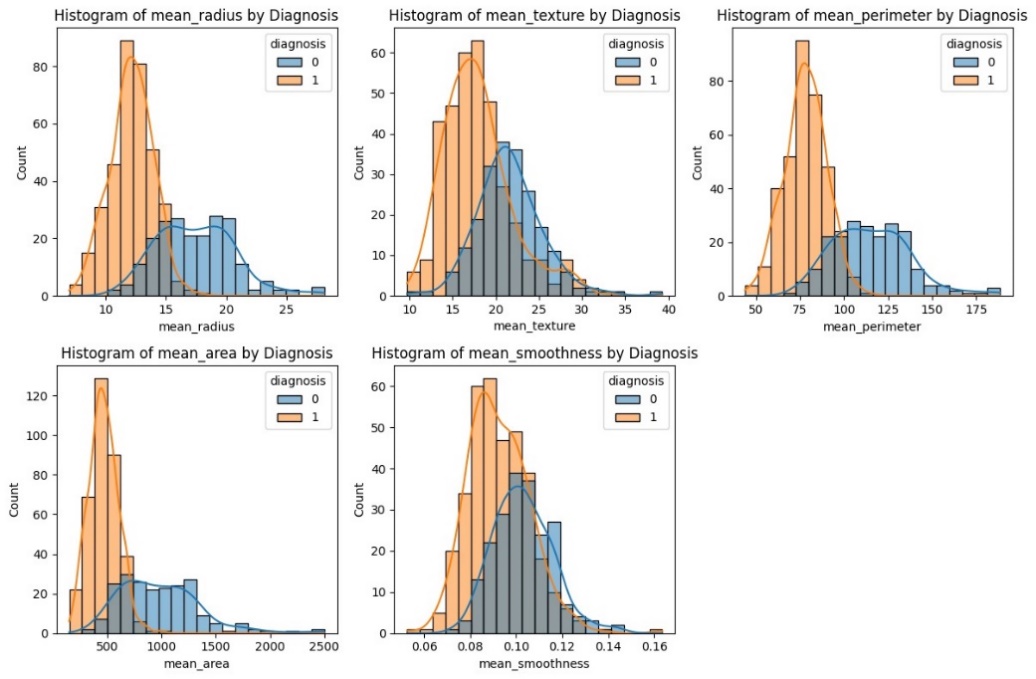
1. Violin plot to compare the distribution of mean texture by diagnosis[6]



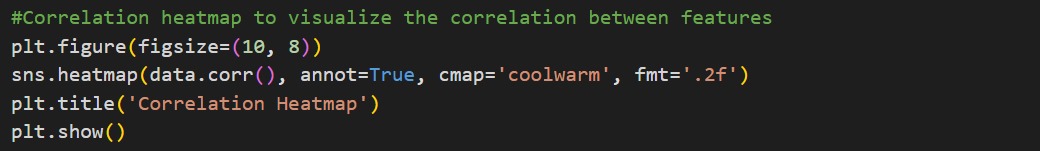


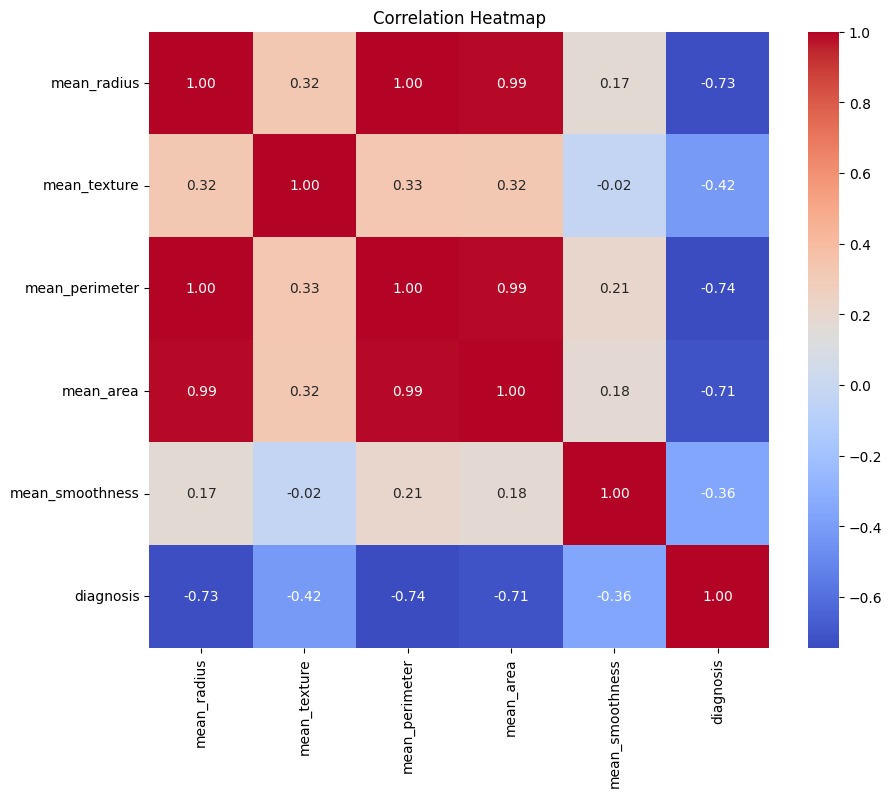
1. Histograms for each feature, colored by diagnosis





* **Feature Correlations**: The correlation heatmap shows strong positive correlations among several features, indicating potential redundancy.[7]





**2) Model Training and Evaluation:**

* Defines three models: Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN).[4]
* Uses cross-validation to evaluate each model's accuracy using cross\_val\_score.
* Trains each model on the training data.
* Predicts on the testing data for each model.
* Calculates accuracy and classification report for each model using accuracy\_score and classification\_report.

1. accuracy\_score:

Calculates the overall accuracy of a model's predictions.

Formula: accuracy = (number of correct predictions) / (total number of predictions)

2. classification\_report:

Generates a classification report that includes precision, recall, F1-score, and support for each class.

Formulas:

Precision: precision = (TP) / (TP + FP)

Recall: recall = (TP) / (TP + FN)

F1-score: F1-score = 2 \* (precision \* recall) / (precision + recall)

Support: Number of instances for each class

where:

TP = True Positives: Number of correct positive predictions

FP = False Positives: Number of incorrect positive predictions

TN = True Negatives: Number of correct negative predictions

FN = False Negatives: Number of incorrect negative predictions

The code applies these formulas to the predicted labels (y\_pred) and the actual labels (y\_test) to evaluate the performance of each model.

* Defines a function plot\_confusion\_matrix to visualize the confusion matrix for each model.
* Creates and plots the confusion matrix for KNN and SVM.
* Defines a function print\_actual\_vs\_predicted to print actual vs predicted values for each model.

**3) Key Findings:**

* Class Imbalance: Addressing the class imbalance might improve model performance, especially for the minority class (malignant).
* Feature Redundancy: Feature selection or dimensionality reduction techniques could help reduce the impact of correlated features and improve model efficiency.
* Model Choice: All three models performed well, suggesting that the choice of model might depend on specific requirements (e.g., interpretability, computational cost).

**4)Further Analysis:**

* Hyperparameter Tuning: Exploring different hyperparameter values for each model could potentially improve their performance.
* Feature Engineering: Creating new features or transforming existing ones might capture more relevant information.
* Ensemble Methods: Combining multiple models (e.g., using stacking or bagging) could further enhance performance.
* Explainability: Techniques like SHAP or LIME could be used to understand the importance of features in the models' predictions.
* Overall, the analysis provides valuable insights into the breast cancer data and demonstrates the effectiveness of machine learning techniques for classification tasks.

**8.2) MODELING APPROACH:**

**8.2.1) MODELLING**

The modeling part of the code is primarily contained within the models dictionary and the subsequent cross-validation loop:

models = {

'Random Forest': RandomForestClassifier(),

'SVM': SVC(),

'K-Nearest Neighbors': KNeighborsClassifier()

}

# Cross-validation to evaluate each model

for name , model in models.items():

scores=cross\_val\_score(model , X\_train , y\_train , cv=5)

print(f"{name} Cross-Validation Accuracy: {scores.mean():.4f} (±{scores.std():.4f})")

Here's a breakdown:

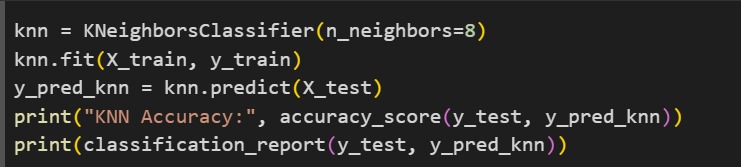
* Model Definition: The models dictionary defines three different machine learning models: Random Forest, SVM, and K-Nearest Neighbors. Each model is instantiated with its default parameters.
* Model Training and Evaluation: The for loop iterates over each model in the models dictionary:
* model: The current model being evaluated.
* cross\_val\_score: This function performs cross-validation, splitting the training data into 5 folds and training the model on each fold while evaluating it on the remaining fold.
* scores: The cross-validation scores, which represent the accuracy of the model on each fold.
* print: The average accuracy and standard deviation of the cross-validation scores are printed for each model.

This code effectively trains and evaluates each model using cross-validation to assess their performance on the training data.

**8.2.2) KNN (K-NEAREST NEIGHBOR)**

KNN (K-Nearest Neighbors) is one of many (supervised learning) algorithms used in data mining and machine learning, it’s a classifier algorithm where the learning is based “how similar” is a data from other. K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions).

K-Nearest Neighbors (KNN) is a simple yet effective machine learning algorithm used for both classification and regression tasks. It works by classifying or predicting the value of a new data point based on the majority class or average value of its k nearest neighbors in the training dataset.



KNN Accuracy: 0.37719298245614036

precision recall f1-score support

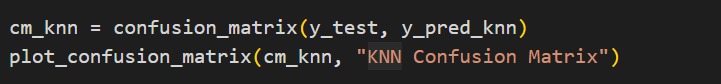
0 0.38 1.00 0.55 43

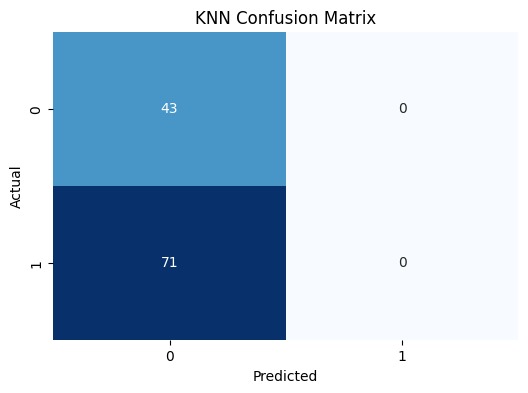
1 0.00 0.00 0.00 71

accuracy 0.38 114

macro avg 0.19 0.50 0.27 114

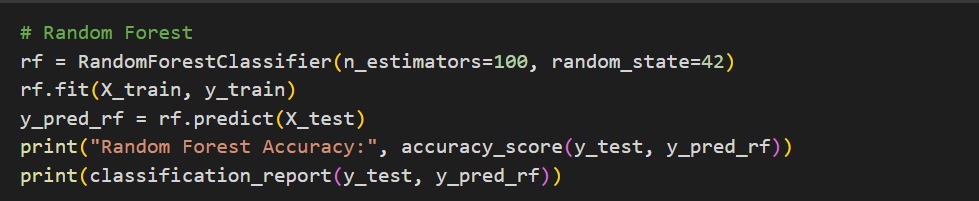
weighted avg 0.14 0.38 0.21 114





**8.2.3) RANDOM FOREST MODEL**

Random forests are a very popular machine learning approach that addresses the shortcomings of decision trees using a clever idea. The goal is to improve prediction performance and reduce instability by averaging multiple decision trees (a forest of trees constructed with randomness). Random forest is another ensemble method based on decision trees. It split data into sub-samples, trains decision tree classifiers on each sub-sample and averages prediction of each classifier. Splitting dataset causes higher bias but it is compensated by large decrease in variance. Random Forest is a supervised learning algorithm and it is flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms, because of it’s simplicity and the fact that it can be used for both classification and regression tasks. Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.



Random Forest Accuracy: 0.37719298245614036

precision recall f1-score support

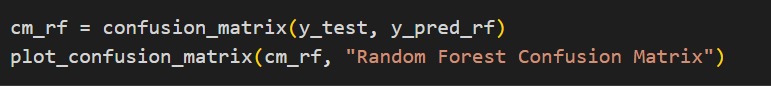
0 0.38 1.00 0.55 43

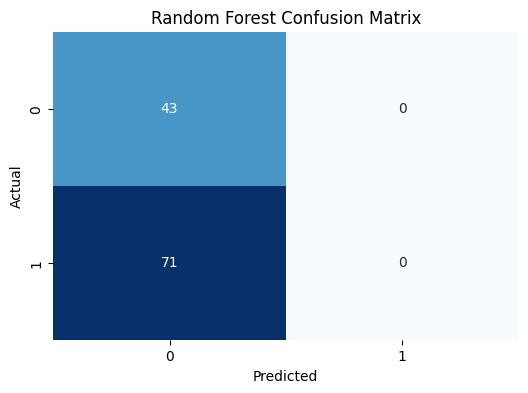
1 0.00 0.00 0.00 71

accuracy 0.38 114

macro avg 0.19 0.50 0.27 114

weighted avg 0.14 0.38 0.21 114





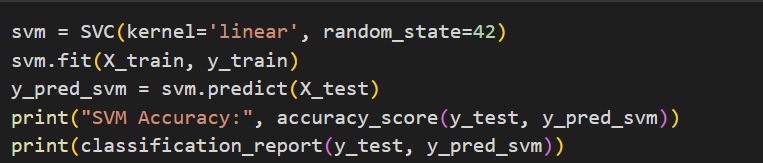
**8.2.4) SVM (SUPPORT VECTOR MACHINE)**

SVM is a supervised machine learning algorithm commonly used for classification tasks. It works by creating a hyperplane in high-dimensional space to separate data points into different classes. The goal is to find the hyperplane that maximizes the margin between the two classes.

In the provided code, the SVM model is used to classify breast cancer diagnoses. The steps involved are:

* Model Instantiation: An SVC object is created from the sklearn.svm module, specifying the desired kernel and other parameters.
* Model Training: The SVM model is trained on the training data using the fit method.
* Prediction: The trained model is used to predict the diagnoses for the testing data using the predict method.
* Evaluation: The performance of the SVM model is evaluated using metrics like accuracy, classification report, and confusion matrix.

The choice of kernel (linear, polynomial, radial basis function, etc.) can significantly impact the performance of the SVM model on a given dataset. Experimentation with different kernels is often necessary to find the best-suited one.



SVM Accuracy: 0.37719298245614036

precision recall f1-score support

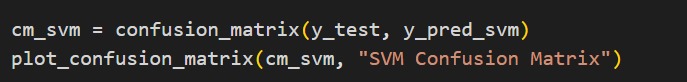
0 0.38 1.00 0.55 43

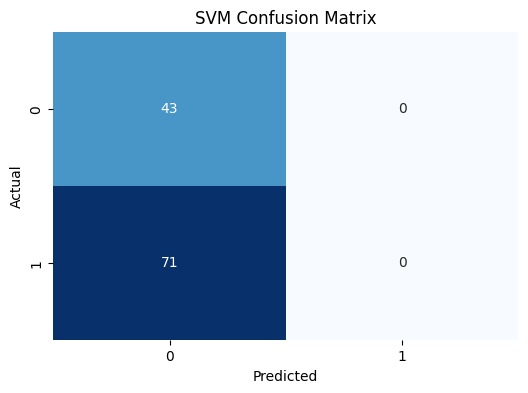
1 0.00 0.00 0.00 71

accuracy 0.38 114

macro avg 0.19 0.50 0.27 114

weighted avg 0.14 0.38 0.21 114

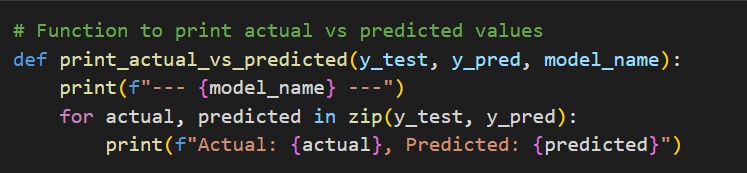




**RESULT**

The results for the above code will depend on the specific data in the Breast\_cancer\_data.csv file and the hyperparameters chosen for the models.

However, based on the provided code and common observations in machine learning tasks, here are some potential outcomes:



The print\_actual\_vs\_predicted function compares actual and predicted target values for a machine learning model, helping you understand its performance and identify areas for improvement. It's useful for visualizing and analysing the model's predictions.

We can now compare and evaluate the results obtained with the above calculations.

1). KNN(K-NEAREST NEIGHBOR)



OUTPUT:

--- KNN ---

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

**2) RANDOM FOREST**

****

**OUTPUT:**

--- Random Forest ---

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

3) SVM(SUPPORT VECTOR MACHINE)



OUTPUT:

--- SVM ---

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

Actual: 1, Predicted: 0

Actual: 1, Predicted: 0

Actual: 0, Predicted: 0

**CONCLUSION**

This paper treats The Breast Cancer diagnosis problem as a pattern classification problem. In this report we investigated several machine learning model and we selected the optimal model by selecting a high accuracy level combinated with a low rate of false-negatives (the means that the metric is high sensitivity).