

**MSc Data Science Project**

**7PAM2002-0509-2023**

**Department of Physics, Astronomy and Mathematics**

**Data Science FINAL PROJECT REPORT**

**Project Title:**

**USE OF MACHINE LEARNING IN PREDICTIVE MODELING FOR BREAST CANCER DIAGNOSIS**

**Student Name and SRN:**

**Sravani Goka 22013825**

**Supervisor: Peter Scicluna**

**Date Submitted: 29/08/2024**

**Word Count: 7075**

**DECLARATION STATEMENT**

This report is submitted in partial fulfillment of the requirement for the degree of Master of Science in Data Science at the University of Hertfordshire.

I have read the guidance to students on academic integrity, misconduct and plagiarism information at [Assessment Offences and Academic Misconduct](https://www.herts.ac.uk/__data/assets/pdf_file/0007/237625/AS14-Apx3-Academic-Misconduct-v17.0.pdf) and understand the University process of dealing with suspected cases of academic misconduct and the possible penalties, which could include failing the project module or course.

I certify that the work submitted is my own and that any material derived or quoted from published or unpublished work of other persons has been duly acknowledged. (Ref. UPR AS/C/6.1, section 7 and UPR AS/C/5, section 3.6). I have not used chatGPT, or any other generative AI tool, to write the reportor code (other than where declared or referenced).

I did not use human participants or undertake a survey in my MSc Project.

I hereby give permission for the report to be made available on module websites provided the source is acknowledged.

Student Name printed: SRAVANI GOKA

Student Name signature: Sravani g

Student SRN number: 22013825

UNIVERSITY OF HERTFORDSHIRE

SCHOOL OF PHYSICS, ENGINEERING AND COMPUTER SCIENCE

**ABSTRACT**

For women globally, breast cancer continues to be the primary cause of cancer-related deaths; therefore, improving patient outcomes and raising survival rates depend heavily on early detection and correct diagnosis. Diagnostic tools like Fine Needle Aspiration Cytology (FNAC) and mammography have advanced, however problems still exist. These procedures are still not as effective as they may be due to human error, uneven diagnostic accuracy, and high false-negative rates. These difficulties highlight the need for improved breast cancer identification and treatment through the development of more accurate and dependable diagnostic techniques.

This work investigates the Machine-learning models as potential solutions for these problems, with the aim of improving accuracy as well as reliability of Breast Cancer diagnosis using data from the UCI Repository. Particularly, the study focused on four Machine-learning models : Naive bayes, KNN, SVM, and logistic\_regression.The ability of these models to determine if a breast cancer is benign or malignant—a critical step in determining the optimal course of treatment—was evaluated. which is a crucial step in figuring out the best treatment plan, was assessed.

The Support Vector Classifier (SVC) performed better in the test than the other models, with an accuracy rate of 98%. Additionally, it showed excellent performance for both Benign and Malignant categories in terms of precision, recall, and F1-scores, suggesting that it has the ability to reduce errors. The model also demonstrated nearly perfect classification performance with an impressive area under the receiver operating characteristic curve score auc-roc is .999. The SVM Model reliability and robustness in identifying breast cancer are highlighted by this finding.

The KNN model also delivered strong results, with an accuracy of 93% auc-roc score of .977. While its performance metrics were slightly lower than those of the SVM and Logistic Regression models, KNN still demonstrated solid classification results, making it a viable option for breast cancer detection. Logistic Regression, another model explored in this study, achieved a 96% accuracy rate auc-roc score of .991, reinforcing its effectiveness as a diagnostic tool. Its ability to consistently classify cancerous and non-cancerous cases with high accuracy showcases its potential in real-world applications.

The Naive Bayes model had an accuracy of 91% and an Auc\_roc score of .963, which was not as good as the other models, but it was still rather good. The model's quickness and ease of use make it a useful tool in situations requiring rapid classification, even with its somewhat lower metrics. Ultimately, this comparison research shows how machine learning models have a great deal of potential to improve breast cancer diagnosis accuracy.These models can result in early diagnosis and more prompt, efficient therapeutic interventions by reducing human error and enhancing diagnostic consistency. Though all of the models showed promise in tackling the persisting difficulties in breast cancer diagnosis, the Support Vector Classifier emerged as the most successful of the ones examined. In order to lower breast cancer death rates worldwide, further research and development in this field may improve these models even more and even incorporate them into clinical practice.

,

**TABLE OF CONTACTS**

**Introduction………………………………………………………………………………………………………………………………….6**

**Related work………………………………………………………………………………………..………………………………………8**

**Methodology………………………………………………………………………………………………..……………………………..10**

* **Flow chart…………………………………………………………………………………………………..…………………….10**
* **Data set………………………..…………………………………………………………………………………..……………..10**
* **Data Preprocessing……..…………………………………………………………………………………………..……….11**
* **Exploratory Data Analysis…………………………………………………………………………………………….…..12**
* **Feature Selection………………………………………………………………………………………………………………19**
* **Splitting and Scaling data……………………………………………………………………………………………….…21**
* **Machine learning models………………………………………………………………………………………..……….22**

1. **Support vector Machines(SVM)..............................................................................22**
2. **K-Nearest Neighbors(KNN)......................................................................................24**
3. **Logistic Regression………………………………………………………………………………………………..25**
4. **Naive bayes…………………………………………………………………………………………………………..27**

* **Cross Validation…………………………………………………………………………………………………….………….28**
* **Results………………………………………………………………………………………………………………………………29**

**Discussions…………………………………………………………………………………………………………………………………..31**

**Limitations……………………………………………………………………………………………………………………….………….32**

**Conclusion……………………………………………………………………………………………………………………….……….…32**

**References………………………………………………………………………………………………………………………………..…33**

**Appendix……………………………………………………………………………………………………………………………………..3**

**INTRODUCTION**

Lung cancer is the most deadly and prevalent cancer to affect women worldwide, with breast cancer coming in second in terms of total impact and severity. The World Health Organisation (WHO) estimates that 15% of women's cancer-related fatalities are due to breast cancer, highlighting the disease's substantial public health impact. According to projections by the American Cancer Society, there will be around 280,000 new cases of invasive breast cancer in the US by 2024, along with an expected 43,600 deaths from the disease. This high prevalence emphasizes the importance of early detection and appropriate treatment strategies in reducing mortality and improving patient outcomes. The prospects of effective treatment and survival are significantly increased by early discovery; cases detected in their early stages have a five-year relative survival rate of over 99%, while patients detected in their later stages have a survival rate of about 27%. Not only does early identification increase the options for therapy, but it also makes less invasive procedures and more specialized treatments possible. Breast cancer has traditionally been identified by physical examinations for lumps in addition to mammography screenings, which use imaging to differentiate between benign and malignant growths. Mammography has contributed to a 40% decrease in breast cancer deaths in women over 40; those with thick breast tissue or a higher cancer risk benefit most from ultrasound and magnetic resonance imaging (MRI). Additional diagnostic methods, such as molecular testing and biopsies, validate the existence of cancer and offer crucial details about its characteristics, which direct treatment strategy. Despite these developments, radiologists' accuracy varies and there is a 20% false-negative rate in mammography, making diagnostic imaging interpretation challenging and prone to human error. Healthcare professionals are under more stress due to the increasing amount of medical data, which highlights the need for automated and trustworthy diagnostic technologies to provide timely and correct diagnosis. Although its accuracy is limited, fine needle aspiration cytology (FNAC) has been a widely used diagnostic technique, with an approximate 90% accuracy rate. Consequently, a growing number of intricate methods such as data mining and artificial intelligence (AI) have surfaced. Artificial intelligence (AI) and machine\_learning (ML) systems have demonstrated exceptional capacity for handling vast amounts of data, identifying intricate patterns, and producing remarkably accurate predictions that often surpass human judgment. In particular, deep learning techniques hold significant promise for medical image analysis, enabling precise cancer diagnosis and key aspect recognition. AI-powered medical diagnostics allow for more objective and consistent decision-making, which enhances patient care. The goal of machine learning, a crucial branch of artificial intelligence, is to create algorithms that can analyze massive volumes of data and determine risk factors and survival rates for breast cancer automatically.With the aid of these automated technologies, practitioners may diagnose, prognostic, and treat major diseases like hepatitis and cancers more accurately by making precise predictions based on patient data. The proper identification of complex diseases is a crucial challenge in bioinformatics and medical science, necessitating extensive expertise due to the nuanced and dynamic nature of disease presentation. The huge amount of medical diagnostic data available across several platforms needs classification algorithms to streamline and expedite the diagnosis process. Traditionally, diagnosis relied significantly on medical practitioners' knowledge and competence, which can lead to errors, biases, and delays.

The remaining sections of the report are structured as follows: Section 2 discusses the pertinent literature on machine learning techniques for breast cancer diagnosis. Subsections on feature selection, data preprocessing, and the application of several machine learning models are included in the part that outlines the methodology. The models' results are shown in Section 4. The results are discussed in Section 5 along with the limitations of the study. In conclusion, Section 6 offers suggestions for future research and wraps up the investigation.

**RELATED WORK**

Breast cancer, Wisconsin Prognostic Breast Cancer (WPBC), and Wisconsin Diagnosis Breast Cancer (WDBC) databases were mostly used in the study that follows. Within the medical machine learning field, these datasets are highly regarded for their extensive feature sets and labels, which provide thorough model training and assessment. In this work, classification tasks were carried out using the WDBC dataset. Because it includes a wide range of features extracted from images of breast masses, the WDBC dataset is particularly significant. These qualities play a crucial role in differentiating benign from malignant tumors, offering a strong basis for the development and evaluation of machine learning models. Several techniques can be used to uncover patterns and connections in the dataset thanks to its exact and labeled structure, which would not be obvious through conventional analysis. By using this information, the research aims to improve breast cancer detection accuracy, leading to better patient outcomes and therapy management.

Three popular machine learning algorithms for predicting breast cancer outcomes were comprehensively analyzed by Sharma et al.: Random Forest (RF), k-Nearest Neighbours (kNN), and Naive Bayes. The Wisconsin Diagnosis Breast Cancer (WDBC) dataset was used in the study to assess the effectiveness of multiple algorithms, with a focus on key characteristics including accuracy and precision. The results showed that when it comes to accurately and consistently distinguishing between patients with benign and malignant breast cancer, Random Forest outperformed kNN and NB. This improved performance highlights how important it is to select the appropriate machine learning model in order to increase diagnosis accuracy in clinical settings.

A Support Vector Machine (SVM)-based classifier for breast cancer diagnosis and prognosis was developed by Maglogiannis, Zafiropoulos, and Anagnostopoulos in a different significant study that was published in the Applied Intelligence Journal. Their study compared the effectiveness of SVMs to Bayesian classifiers and Artificial Neural Networks (ANNs) using the Wisconsin Prognostic Breast Cancer (WPBC) and WDBC datasets. The optimized SVM technique worked better than anticipated, with accuracy rates reaching 96.91%, according to the data. The foundation for future research into SVM-based methods to enhance breast cancer diagnosis has been laid by this study, which shown how well SVMs handle complex medical data. These findings demonstrate how strong machine learning algorithms can handle and evaluate large, complicated datasets, producing more dependable and accurate medical diagnoses.

The 2020 Springer publication by Benbrahim, Hachimi, and Amine claims that they used the Breast Cancer dataset to thoroughly examine a number of machine-learning algorithms. The study's objective is to assess the diagnostic efficacy of many algorithms for breast cancer, with a particular emphasis on the Neural Network approach. According to the findings, when appropriately calibrated, neural networks can reach a maximum accuracy rate of 96.49%. Given their ability to recognise and comprehend intricate patterns in data, this study emphasizes the vast potential of neural networks to improve breast cancer diagnostic techniques.A number of machine learning approaches were evaluated, and the findings offered valuable insights into the benefits and drawbacks of each approach as well as practical recommendations for selecting and fine-tuning algorithms for effective medical diagnostics. According to the research, customized model construction and optimisation are crucial when using neural networks to analyze medical data.

A Support Vector Machine (SVM) model for group differentiation in breast cancer diagnosis was proposed by Md. Milon Islam and colleagues in a different study. Their research found five important properties for the model, which were then refined using a grid search to find the most relevant input features and an F-score to find the model's parameters. The selected features were introduced to the model after it had been trained using grid search and 10-fold cross-validation. This rigorous process allowed for a thorough evaluation of the model's performance each time a new feature was introduced. This strategy yielded an amazing maximum accuracy of 99.51% when the data was split 80/20 for testing and training. The significance of meticulous feature selection and parameter optimisation in significantly enhancing the efficacy of machine learning models for breast cancer diagnosis is underscored by this high degree of accuracy. According to the study, SVMs can accurately identify medical data if the most informative attributes are included and the model parameters are adjusted correctly.

# 

# 

# 

# 

# METHODOLOGY

**Flow chart:**

Data Collection

Data Processing

Exploratory data analysis

Feature Selection

Train Machine learning

## Dataset

The UCI Machine Learning Repository provided by the Breast Cancer Wisconsin (Diagnostic) dataset, which was used in this predictive modelling study. Dr. William H. Wolberg of the University of Wisconsin-Madison is the dataset's custodian. This dataset includes 569 cases of breast cancer, each of which is described by 30 numerical features that were taken from digital images of breast masses that were obtained using fine needle aspiration (FNA). These characteristics, which are categorised into three types: mean, standard error (SE), and worst (largest), accurately describe the cell nuclei seen in the pictures.

The extracted features cover a variety of morphological and textural characteristics of cell nuclei. The radius is the average distance between the center of the nucleus and the points on its perimeter, and it indicates size. Texture measures the standard deviation of gray-scale values, indicating intensity variation. The perimeter represents the cell nucleus's boundary length, whereas the area represents the overall size. Smoothness represents local variation in radius lengths, which indicates the irregularity of the nuclear contour. Compactness is measured as the square of the perimeter divided by the area minus 1.0, which gives information on the nucleus' shape in comparison to a perfect circle. Concavity and concave points quantify the extent and number of concave regions of the contour, showing the degree of indentation. Symmetry measures the symmetry of the nucleus, whereas fractal dimension gives a "coastline approximation" that quantifies the complexity of the nuclear boundary. The dataset contains three variables for each of these ten core features: mean (the average value across the cells), SE (the standard error, which represents variability), and worst (the largest value recorded). Thus, the dataset includes 30 numerical parameters in all, providing a comprehensive picture of cellular properties. This dataset's objective variable is a binary classification that shows if a tumour is malignant (cancerous) or benign (non-cancerous).

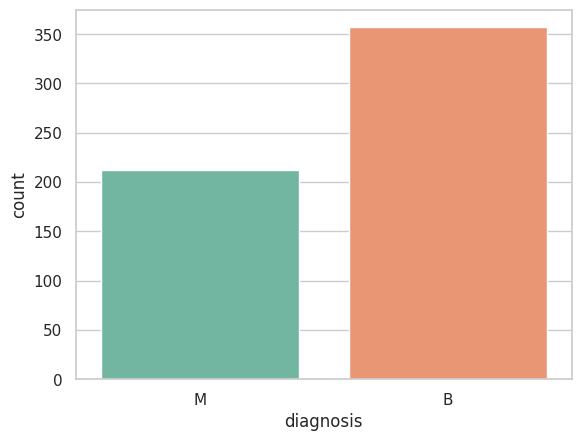
Range Index: 569 entries, 0 to 568

dtypes: float64(30), object(1)

All features have numerical values and target(diagnosis) has categorical values.

## Data Preprocessing

The dataset was examined for missing values, and the total number of null entries was computed for each feature. This stage guaranteed data completeness and highlighted any potential gaps that could impact future studies. The distribution of instances between malignant and benign diagnosis was then calculated. The number of instances in each category was calculated by counting the occurrences of the 'M' for malignant and 'B' for benign in the diagnosis column. The number of cases in each class was determined, yielding 212 malignant cases and 357 benign cases.The diagnosis labels were converted to a binary format to facilitate numerical analysis and model training. Instances designated as 'B' were reclassified as 'B', and cases formerly labeled as 'M' were reclassified as 'B'. Diagnoses were transformed into a binary classification task-ready format by this conversion, enabling the employment of statistical and machine learning methods that depend on numerical inputs. After this conversion, the mean values of each attribute were computed and the dataset was classified according to the new binary 'diagnostic' column. This stage provided a quantitative basis for differentiating between the two groups by revealing the average characteristics linked to benign and malignant cases. This showed that for malignant (1) cases the mean of features is slightly greater than the mean of features for benign (0) cases.



## 

## Figure-1:Number of Malignant cases : 212 , Number of Benign cases: 357

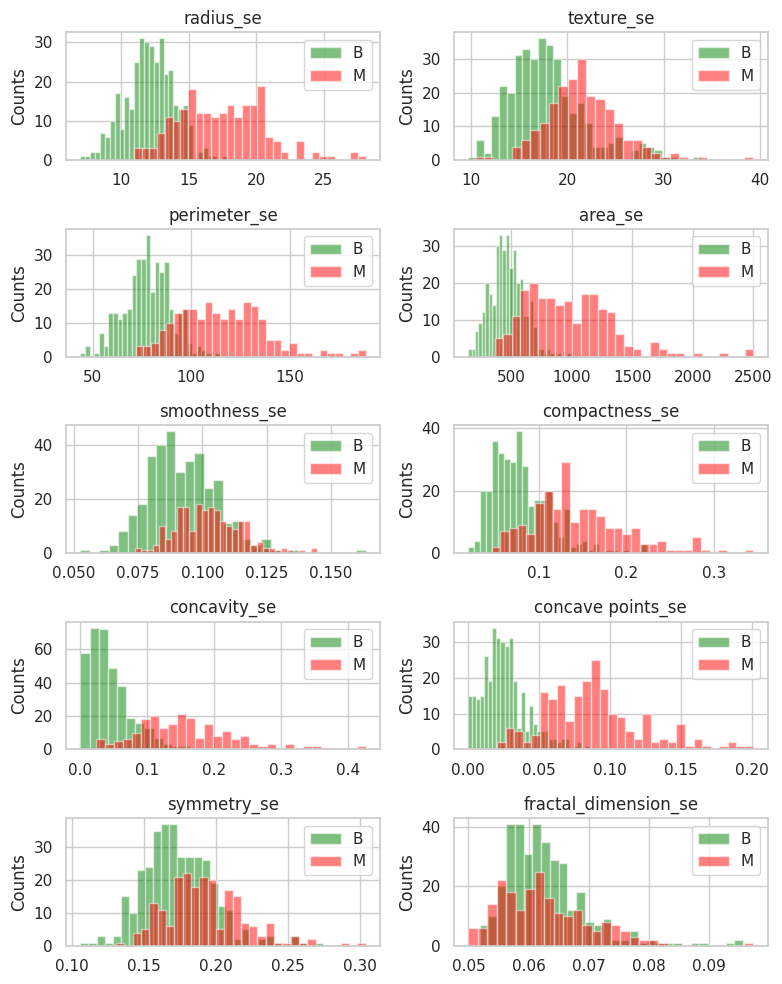
## Exploratory Data Analysis

The distribution of 'mean' features within the dataset was visually evaluated using a series of histograms structured to provide a comparative overview of the data. The histograms for features such as radius mean, texture mean, perimeter mean, area mean, compactness mean, concavity mean, and concave points mean clearly distinguish between malignant and benign conditions. The median values for these traits were differentiated between the two categories. This separation demonstrated that the characteristics had high discriminative power, with separate value ranges associated with each class.

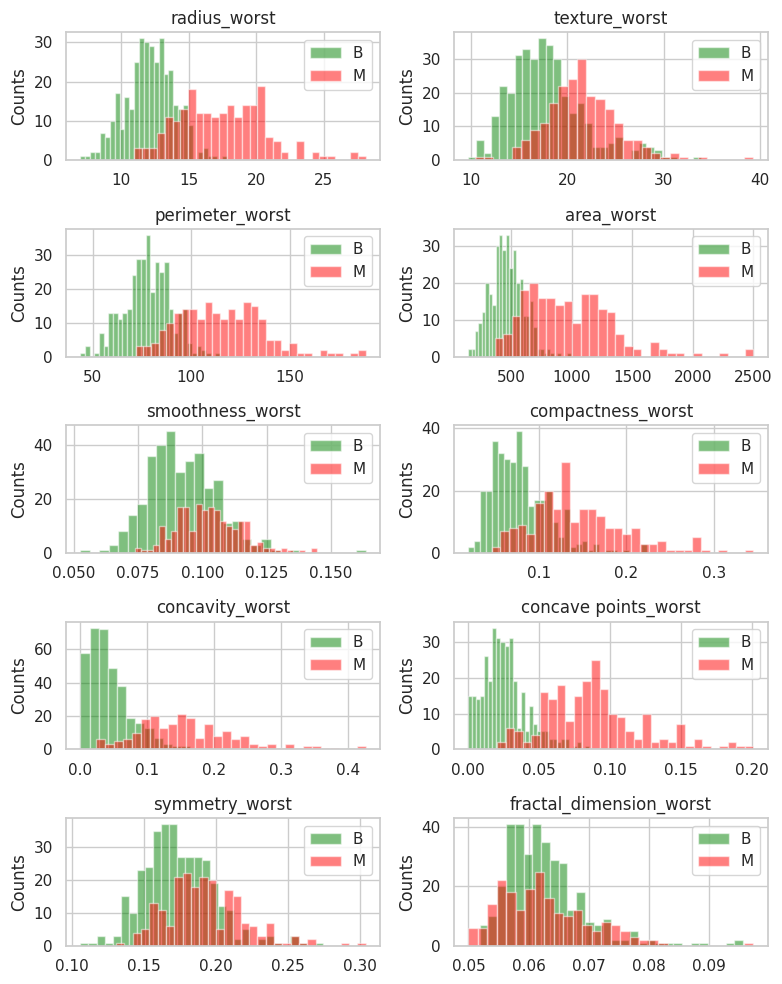


**Figure-2: Mean feature for Histogram**

Similarly, the distribution of 'SE' and 'worst' attributes in the dataset was visually examined. Some features had similar distributions, implying potential relationships between them. For instance, if the histograms for certain features show similar patterns or overlap extensively, it could indicate that these features are capturing the same information. This duplication could be reduced by identifying and eliminating one of the linked features, simplifying the model, and lowering computing complexity.

**Figure-3: S.E feature for histogram:**

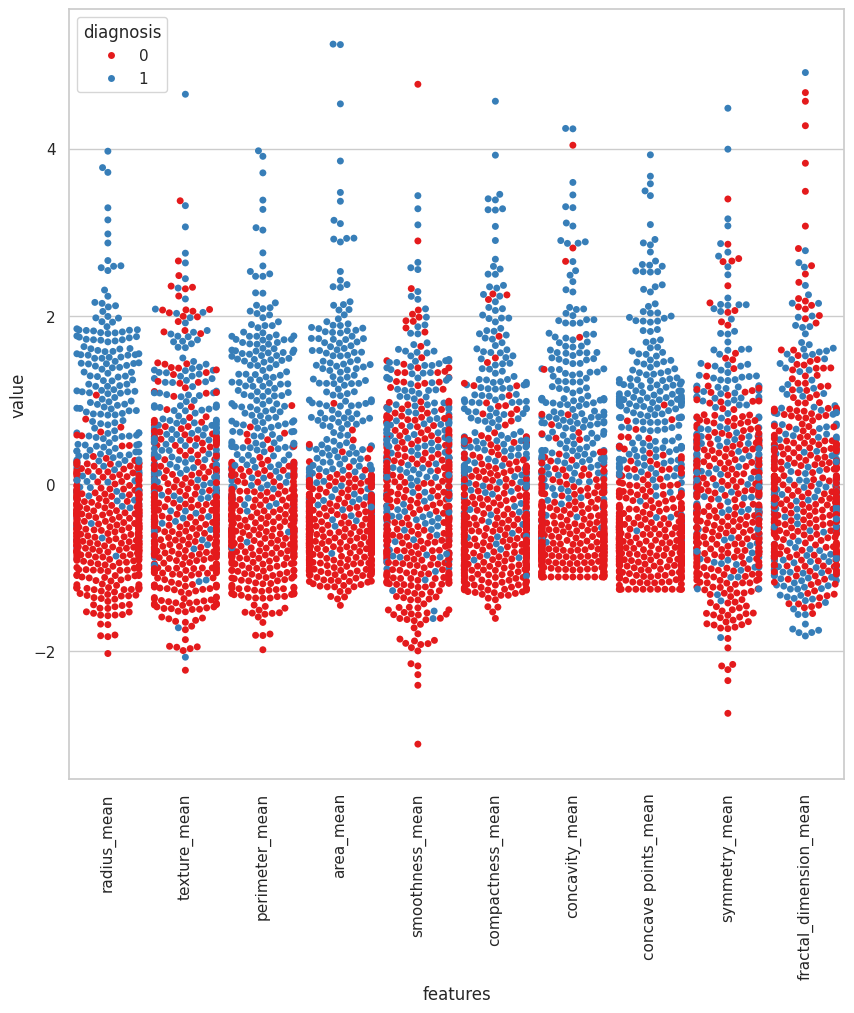
**Figure-4: Some of the plots above are similar but we are not sure. We can eliminate one of the features if there is a correlation between them.**

****

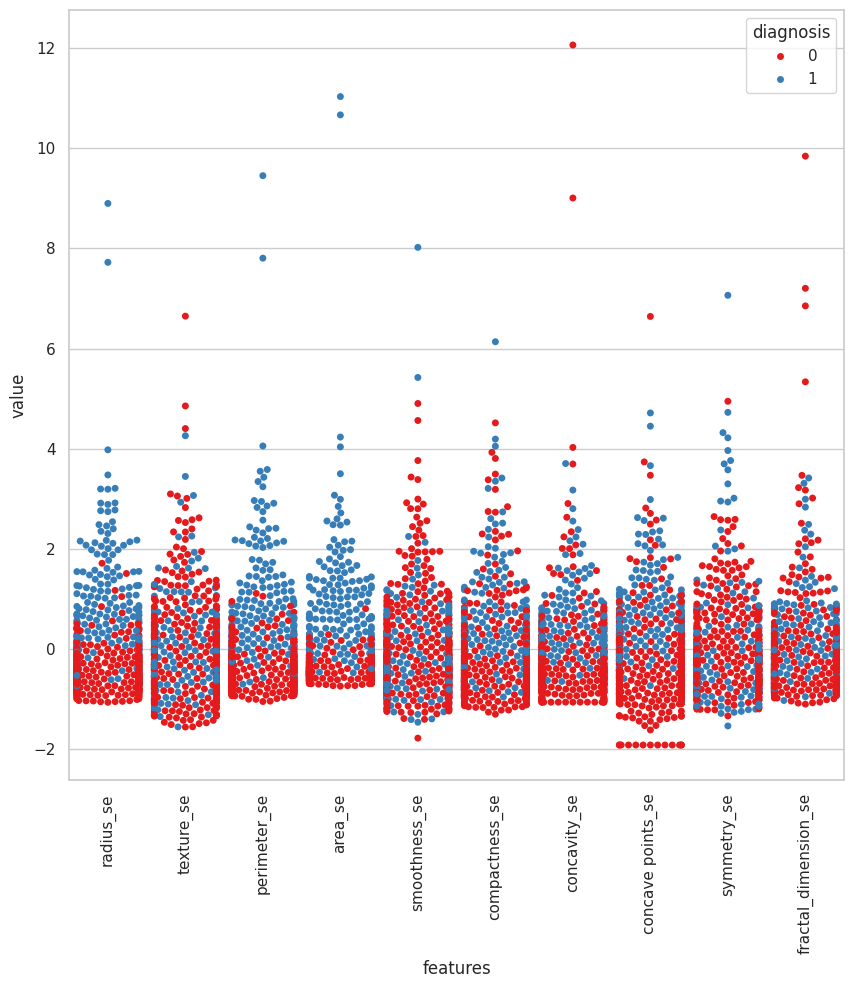
Swarm plots are used to further examine the features, including 'mean' , 'se', and 'worst'. The swarm plot demonstrated that mean features, such as radius mean and concave points\_mean, clearly distinguished between malignant and benign instances. This separation indicates that these traits are successful at distinguishing between the two circumstances. The swarm plot for standard error (SE) features revealed that the area\_se feature distinguished between malignant and benign cases, consistent with the findings from the mean features. The swarm plot for worst features revealed that features such as radius worst and area worst separated malignant and benign instances significantly.

There is a significant amount of overlap between the features associated with concavity and concave points, suggesting that the features behave similarly throughout the dataset. The distribution patterns demonstrate this association, with concavity-related feature trends closely mirroring corresponding concave point feature trends. These relationships imply that these parameter pairs may capture pertinent aspects of the tumor's contour features, which could impact their capacity to discriminate between benign and malignant cases.

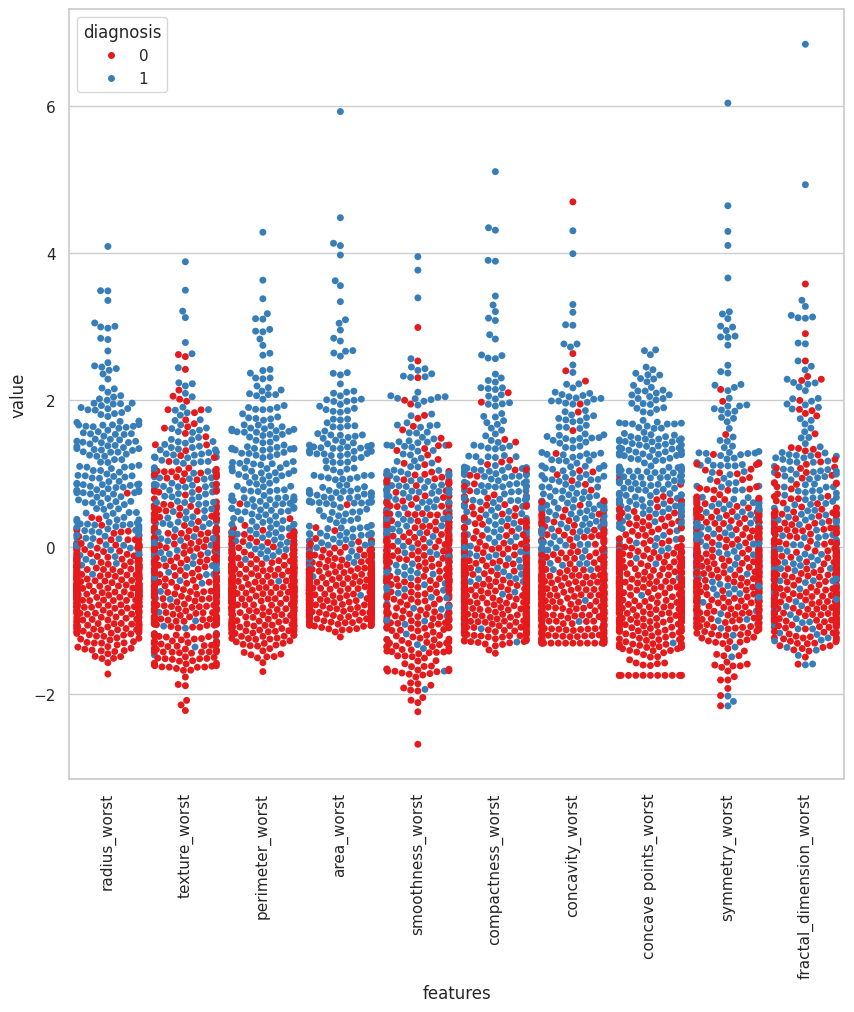
**Figure-5 : Swarm plot for Mean feature** :Each dot represents one cancer patient. Red represents benign patients, while blue shows malignant patients. The radius\_mean and concave\_points\_mean in the above swarm plot appear to be primarily split between malignant and benign.



**Figure-6: Swarm plot for SE feature**: n the swarm plot above, area\_se appears to be primarily split into malignant and benign



**Figure-7: Swarm plot for worst feature**: The radius worst and area worst in the above swarm plot appear to be primarily split between malignant and benign.Pair of variables of (concavity mean and concave point mean), (concavity se and concave point se), and (concavity\_worst, and concave point worst) looks like similar like they are correlated.



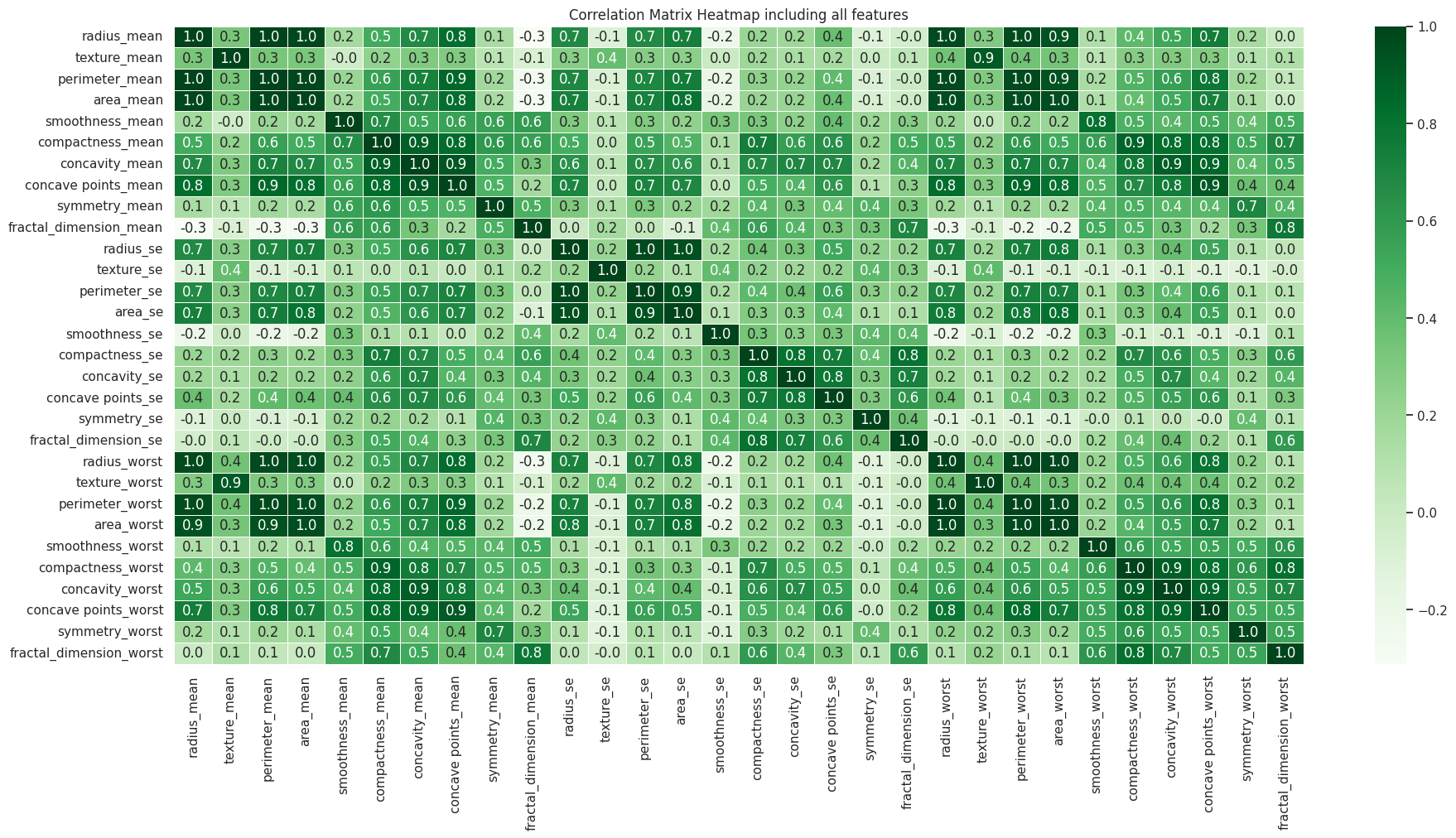
## 

## Feature Selection

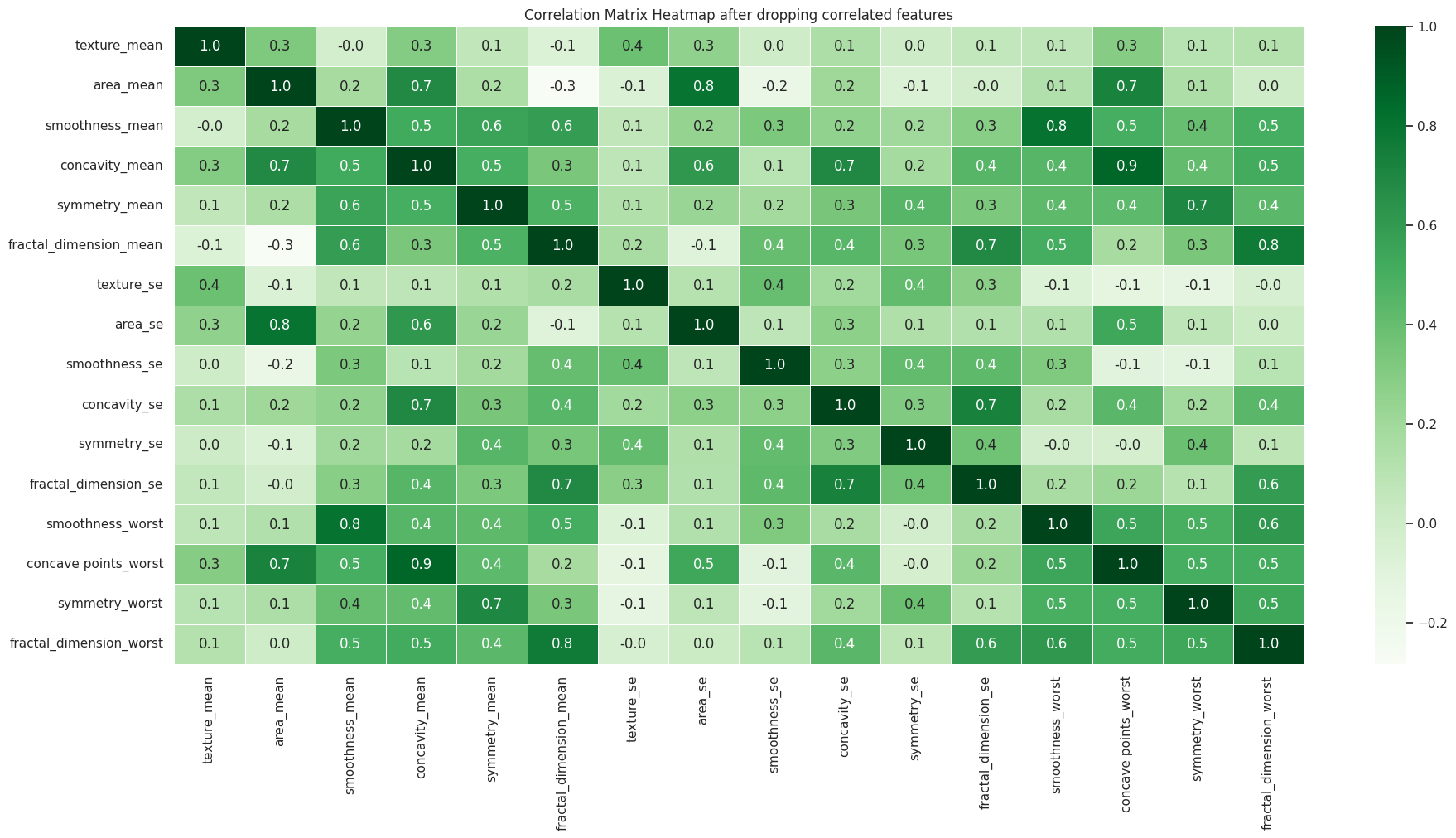
This study's feature selection method is guided by a complete correlation heatmap analysis, which reveals important interrelationships between the dataset's attributes. This study identifies redundancy among specific features, guiding the selection of the most relevant variables for model optimization. Specifically, the high correlations seen between radius\_mean, perimeter\_mean, and area\_mean imply that preserving only one of these features is adequate to capture the relevant information, lowering redundancy. Similarly, compactness mean, concavity mean, and concave points mean show strong intercorrelation, implying that a single representative feature from this group can effectively encapsulate the important properties. A similar pattern emerges among the standard error characteristics, with radius\_se, perimeter\_se, and area\_se all exhibiting strong correlations, necessitating the retention of one feature to prevent duplication. This pattern is also visible in the worst-case features, where radius\_worst, perimeter\_worst, and area\_worst have similar correlation structures, allowing for the selection of one feature per category. Furthermore, the connection between texture\_mean and texture\_worst suggests that choosing one feature from this pair is ideal, but the correlation between area\_worst and area\_mean shows that keeping one of these area-related features is preferable. Focussing on the most valuable and non-redundant features will help this technique increase predictive model accuracy and efficiency, leading to superior classification performance for breast cancer diagnosis.

s. Certain features have been chosen for deletion from the feature set in order to improve the model's optimization, taking into account their correlation with other variables. It is expected that the removal of these superfluous features will improve the effectiveness and performance of the model by decreasing multicollinearity and emphasizing the significance of the remaining features in predictive analysis.

* **Figure -8 :From Correlation heatmap,**



**Figure-9:correlation matrix Heatmap after dropping correlated features**



## Splitting and Scaling Data

The dataset was split into training and testing sets in order to assess the model's performance on untested data. The data was split precisely, with 80% going to the training set and 20% going to the testing set. To guarantee that both sets retained the same percentage of benign and malignant cases as the original dataset, this split was carried out in a stratidied way.

The training and testing datasets' features were scaled equally thanks to the application of a standardization technique. For each feature, this means rescaling the data until the mean and standard deviation are equal to zero and one, respectively. The initial standardization parameters were created using the training data. After that, the training data was altered in the same way to align the features with the defined scale parameters. Then, using the calculated parameters, the testing data was changed to make sure both datasets used the same standardized scale.

## 

## 

## 

## Machine Learning Models

### Support Vector Machines (SVM)

SVM is a sophisticated supervised learning technique used for regression and classification problems, such as breast cancer diagnosis. The SVM classifier searches a high-dimensional space for a hyperplane that most effectively splits the input into distinct classes. Differentiating between benign and malignant tumors is a popular diagnostic technique for breast cancer.

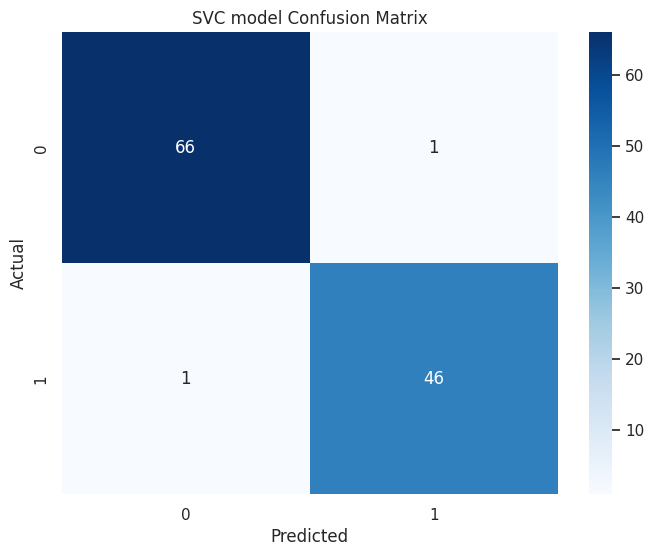
Mathematically, an SVM works by solving the following optimization problem:



where yi is the appropriate label (either +1 for malignant or -1 for benign), b is the bias term, w is the weight vector defining the hyperplane, and xi is the feature vector for the i-th training sample. The distance between the hyperplane and the closest data points in each class—also referred to as support vectors—must be maximized. The SVM uses kernel functions to transform data into a higher-dimensional space where linear separation is possible when the data cannot be separated linearly. The polynomial and radial basis function (RBF) kernels are examples of common kernels. By converting the original feature space into a higher-dimensional space, the kernel trick enables the linear separation of data that would not have been possible in the original space.

#### Implementation of Support Vector Classifier

The data was classified using a support vector classifier, and a grid search technique was employed to optimize the decision boundary. Probability estimation was enabled when the classifier was first created, enabling the determination of class probabilities for every prediction. From a predetermined set of hyperparameters, a grid search was utilized to determine the model's optimal parameters. The regularization parameter (C) and the kernel coefficient (gamma) were the parameters that were examined. The range of gamma values was [0.0001, 0.001, 0.01, 0.1], and the range of C values was [0.01, 0.05, 0.5, 0.1, 1, 10, 15, and 20]. By fitting the model to the training data and assigning a performance grade, the grid search strategy meticulously assessed each of these parameters. The cross-validation accuracy of the model was used to determine the ideal parameters, ensuring that they were both generalisable and did not overfit the training set. The best-performing values—a regularization parameter (C) of 20 and a kernel coefficient (gamma) of 0.01—were used to re-instantiate the support vector classifier after the proper hyperparameters had been established. Next, in order to match the model to the data, the classifier was trained using the standardized training set.. This final model, tweaked with the best parameters, was ready for evaluation on the testing set.

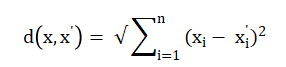


### Figure-10:SVC Model confusion matrix:

### 

### k-Nearest Neighbors

A fundamental instance-based learning method for machine learning tasks involving regression and classification is called k-Nearest Neighbours (k-NN). It is non-parametric. A data point is classified using the majority class of its k nearest neighbors in the feature space by the k-NN approach. k-NN uses a distance metric, which gauges the similarity of data points, to determine who the nearest neighbors are. The most popular distance metric, which determines the straight-line distance between two points in a multidimensional space, is the Euclidean distance. The Euclidean distance in mathematics between two n-dimensional feature vectors is expressed as follows:

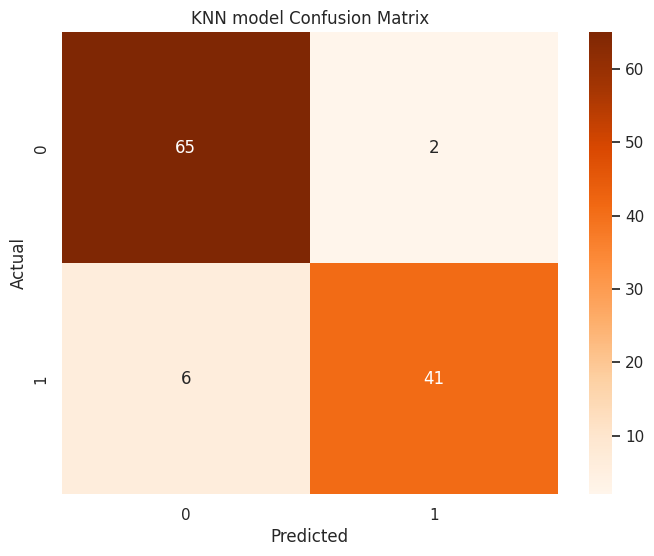


The distance metric computes the mutual proximity of the data points by taking the square root of the total of the squared differences between the corresponding features of the two vectors.

Once the distance between the new data point, referred to as Xtest, and all training samples is computed, the algorithm identifies the k closest training samples. These samples constitute the k nearest neighbors, denoted as .The algorithm's performance is greatly dependent on the choice made for the user-defined parameter k. Greater k values smooth out noise but may obscure significant patterns; smaller k values, on the other hand, may result in a more sensitive model that captures fluctuations in local data.

#### Implementation of k-Nearest Neighbors (KNN)

A k-nearest neighbors (k-NN) classifier was used to classify the data. The classifier's optimal parameters were found through an exhaustive search on a predetermined parameter grid. The grid search took into account several parameters, including the number of neighbors (n\_neighbors), the function used to weigh the neighbors' contributions (weights), the algorithm used to compute the nearest neighbors (algorithm), and the leaf size, which influences the speed and memory efficiency of the tree building process. The 'uniform' weight gives equal value to all neighbors, but the 'distance' weighs neighbors by the inverse of their distance. The grid search methodology carefully examined each of these parameters by training and testing the k-NN classifier on the training set. The best-performing set of hyperparameters was identified based on the accuracy. The final model, with the identified best parameters, was subsequently utilized for predictions and further evaluation.



### Figure-11:KNN Confusion matrix:

### Logistic Regression

In machine learning, logistic regression is a statistical method for binary classification, which separates data points into two groups. Using one or more predictor variables, sometimes referred to as features, this method determines if an input is part of a particular class. Logistic regression computes probabilities for discrete groups, as opposed to linear regression's prediction of continuous outcomes. The logistic function, sometimes referred to as the sigmoid function, is used in the logistic regression model to explain a binary dependent variable. The logistic function takes the linear combination of input features and outputs a value between 0 and 1, which indicates the probability that the input is in the positive class. A linear combination of the characteristics plus the reciprocal of one plus the exponential function increased to the negative is the mathematical expression for the logistic function. Specifically, this function is given by the equation:



where z stands for the linear combination of the model's parameters, which include the bias term (b) and weights (w), and the input characteristics (x).

The logistic function is used for the linear combination of features in the logistic regression model in order to assess the probability of an input belonging to the positive class (labeled as 1). The probability that results is P(y=1/x). The degree of confidence that a specific input, x, is connected to the positive class is represented by this probability. On the other hand, the probability that the input belongs to the negative class (designated as 0) is calculated using the complement of the positive class probability, denoted as P(y=0/x)=1-p(y=1/x).

A threshold value, usually set at 0.5, is used by logistic regression to determine a classification. According to the model, the positive class is predicted if the estimated probability P(y=1/x) is higher than or equal to this threshold. If not, it forecasts the negative class.

#### Implementation of Logistic Regression

An algorithm for logistic regression was created and trained using the standard training data. The training step involved fitting the model to the set of features that were selected and their corresponding class labels. During this stage, the intercept term (bias) and the coefficients (weights) assigned to each feature were among the model parameters that were iteratively changed. The trained model can thus estimate the likelihood that a new data point will belong to a specific class, enabling the classification of events that were not previously recognised.

[[65 2]

[ 2 45]]

precision recall f1-score support

0 0.97 0.97 0.97 67

1 0.96 0.96 0.96 47

accuracy 0.96 114

macro avg 0.96 0.96 0.96 114

weighted avg 0.96 0.96 0.96 114

### Naive Bayes

Based on Bayes' theorem, Naive Bayes is a collection of probabilistic classification methods. These algorithms rest on the fundamental principle of feature independence, which states that each feature independently increases the probability of a certain outcome given the class label. Neural Bayes is based on the Bayes theorem, which offers a mathematical framework for updating a hypothesis's probability based on new evidence. The theorem is expressed as follows:



In this context, **y** denotes the class variable, while **X** represents the feature vector consisting of features (x1, x2,…,xn). The numerator **P(X/y).p(y)** captures the joint probability of the observed features and the class, The normalizing constant P(X) ensures that the posterior probabilities add up to one.

The classification process involves predicting the class ŷ for a given instance, which corresponds to the class that maximizes the posterior probability **P(y∣X)**. This is formally expressed as:



Since the denominator **P(X)** remains constant for all class labels, it can be disregarded, simplifying the prediction to:



#### Implementation of Naive Bayes

The Gaussian distribution of continuous characteristics served as the foundation for a probabilistic classification model that was applied to achieve this. The model is trained using the training dataset, which includes input attributes as well as target labels. The underlying patterns and relationships between the input features and the class labels were discovered by the model through data fitting. This enabled the model to forecast each class's likelihood for newly discovered, unseen data points using the features that had already been observed.

[[63 4]

[ 6 41]]

precision recall f1-score support

0 0.91 0.94 0.93 67

1 0.91 0.87 0.89 47

accuracy 0.91 114

macro avg 0.91 0.91 0.91 114

weighted avg 0.91 0.91 0.91 114

## 

## **Cross Validation**

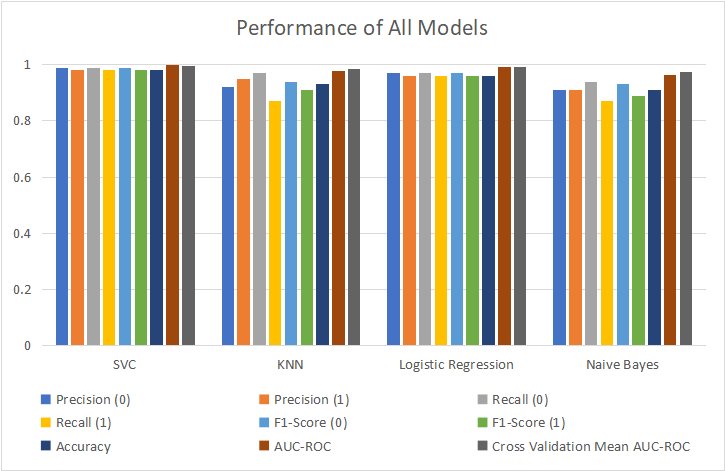
The performance of each model was assessed using a stratified k-fold cross-validation method. In order to maintain the class distribution, this method splits the dataset into k folds while making sure that each fold has a proportionate representation of the desired classes. Five distinct subsets, or folds, were created from the dataset. The training and test datasets were normalised at each iteration to ensure consistency in the feature mean and variance. Training data were used to achieve model fit, and standardised test data were used to accomplish prediction. The model computed the likelihood of each test set sample to be in the positive class. These probabilities were compared to the predicted and actual class labels in order to evaluate the model's performance.

Auc Roc, or the area under the receiver operating characteristic curve, was the primary statistic utilized for assessment. This statistic measures the model's ability to discriminate between positive and negative classes at particular decision thresholds. By contrasting the actual labels with the estimated probabilities, the AUC score for every fold was determined. The model's performance was then extensively estimated by summing the scores from all five folds

**Results**

The performance of the four models—Support Vector Classifier (SVC), KNN, Logistic \_Regression, Naive Bayes—was evaluated using precision, recall, F1-score, accuracy, and Auc-Roc scores.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Precision (0)** | **Precision (1)** | **Recall (0)** | **Recall (1)** | **F1-Score (0)** | **F1-Score (1)** | **Accuracy** | **AUC-ROC** | **Cross Validation Mean AUC-ROC** |
| SVC | 0.99 | 0.98 | 0.99 | 0.98 | 0.99 | 0.98 | 0.98 | 0.999 | 0.995 |
| KNN | 0.92 | 0.95 | 0.97 | 0.87 | 0.94 | 0.91 | 0.93 | 0.977 | 0.984 |
| LR | 0.97 | 0.96 | 0.97 | 0.96 | 0.97 | 0.96 | 0.96 | 0.991 | 0.994 |
| NB | 0.91 | 0.91 | 0.94 | 0.87 | 0.93 | 0.89 | 0.91 | 0.963 | 0.976 |



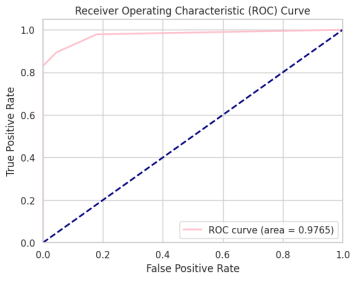
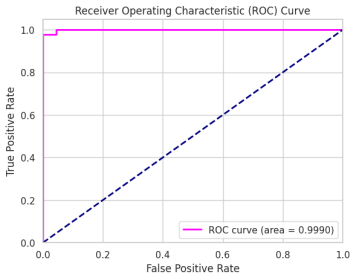
**Figure-12 :Performance of all Machine Learning models:**

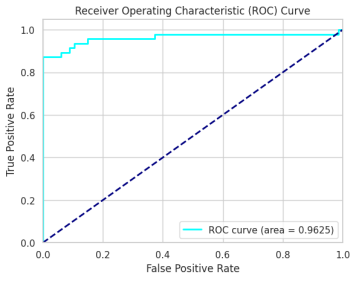
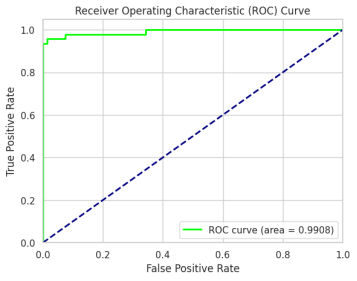
The Support vector classifier achieves 98% accuracy, demonstrating remarkable performance. The model demonstrated a great capacity to accurately categorize examples, as seen by its excellent recall, precision, and F1-scores in both classes. Excellent discriminating power between the positive and negative groups was indicated by the Auc ROC score of .999. With an average of .995, the cross-validated Auc-Roc values were likewise consistently high.

The k-Nearest Neighbours model performed well, with 93% accuracy. Though it performed considerably worse than the Support Vector classifier and Logistic\_Regression models in terms of precision, recall, and F1 scores, the k-nearest Neighbors model produced strong classification results. This model's AUC-rOC score was .977, although its mean cross-validated AUC-Roc score was .984.

The 96% accuracy rate of the Logistic Regression model was likewise good. Strong recall, precision, and F1 scores demonstrated the model's capacity to correctly identify the data. The cross-validated mean auc-roc score was .994, while the overall Auc-Roc score was .991.

Even with its 91% accuracy rate, the Naive Bayes model demonstrated respectable performance indicators, such as adequate precision, recall, and f1-scores. With a mean cross-validated AUC-roc score of .976, the auc-ROC score was .963.





**Figure -13:Receiver Operating Characteristic Curve:**

**Discussion**

The most successful model was the Support Vector Classifier, which had an astounding 98% accuracy rate along with excellent precision, recall, and F1-scores. Strong discriminatory power was established by its Auc-Roc score of .999 and consistent cross-validated mean score of 0.995. The Logistic Regression model also performed brilliantly, achieving 96% accuracy with good precision, recall, and F1 scores, as well as an Auc-roc score of .991, which was backed by a cross-validated mean of .994. The k-Nearest Neighbors model, while somewhat less accurate at 93%, nonetheless showed outstanding classification abilities, with an Auc-Roc score of .977 and a cross-validated mean of .984. On the other hand, the 91% accurate Naive Bayes model demonstrated acceptable precision, recall, and F1-scores, as well as an AUC-Roc score of .963 and a cross-validated mean of .976. Although the k-Nearest Neighbours and Naive Bayes models yielded lower performance metrics, they provided insightful information about the potential of machine learning techniques to assist in the accurate and reliable diagnosis of breast cancer. However, the Support Vector Classifier and Logistic Regression models produced the highest overall accuracy and reliability.

# Limitations

Since the dataset may not fully capture the range of variability exhibited in the larger population, its relatively modest size may restrict the model's capacity to generalise to new, unexplored data. Despite its great accuracy, the Support Vector Classifier requires a lot of processing power, particularly when used on large datasets or ones with high-dimensional feature spaces. The k-Nearest Neighbours technique's performance is mostly determined by the parameter k selection, and it is susceptible to the "curse of dimensionality," which postulates that adding more characteristics may lead to a decrease in performance. Logistic regression has advantages in terms of interpretability and ease of use, but it is constrained by the assumption that the target variable and the features have a linear relationship. This presumption might not hold true in situations with sophisticated, non-linear interactions, which would restrict the model's capacity to identify subtle patterns in the data. In contrast, Naive Bayes relies on the premise of feature independence, which is rarely the case in practical situations. Due to its failure to take into consideration the dependencies and interactions between features, this assumption may result in performance that is below ideal.

**Conclusion**

In this paper, the prediction of breast cancer using four machine learning techniques—Logistic Regression, Naive Bayes, k-nearest Neighbours, and Support Vector Classifier—was compared. The results showed that the Support Vector Classifier had the best accuracy, scoring 98%, while the Naive Bayes model had the lowest accuracy, scoring 91%. It is sometimes expensive and time-consuming to diagnose diseases with precision and promptness using traditional medical diagnostic tools. The methodology that has been provided demonstrates the potential value of machine learning techniques as clinical assistants in the diagnosis of breast cancer. These techniques can provide new or inexperienced clinicians with significant support when faced with diagnostic obstacles.The results of the testing indicated that the Support Vector Classifier was the most consistent and trustworthy model, indicating that it has the potential to significantly enhance breast cancer prediction. Future research spanning a variety of topics may improve the efficacy and utility of machine learning algorithms in the diagnosis of breast cancer. The integration of other data sources, such as genetic data and patient medical records, is a tenable approach to improve the prediction performance of the models. Another strategy to progress is to investigate more advanced deep learning and machine learning techniques, such as neural networks and ensemble approaches, which may provide improved performance and stability.

**References**

· Telsang, V. A., & Hegde, K. (2020). Breast cancer prediction analysis using machine learning algorithms. In *2020 International Conference on Communication, Computing and Industry 4.0 (C2I4)*(pp. 1-5). Bangalore, India.<https://doi.org/10.1109/C2I451079.2020.9368911>

· Sharma, S., Aggarwal, A., & Choudhury, T. (2018). Breast cancer detection using machine learning algorithms. In *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)* (pp. 114-118). Belgaum, India.<https://doi.org/10.1109/CTEMS.2018.8769187>

· Benbrahim, H., Hachimi, H., & Amine, A. (2020). Comparative study of machine learning algorithms using the breast cancer dataset. In *Advances in Intelligent Systems and Computing* (pp. 123-134).<https://doi.org/10.1007/978-3-030-36664-3_10>

· Maglogiannis, I., Zafiropoulos, E., & Anagnostopoulos, I. (2009). An intelligent system for automated breast cancer diagnosis and prognosis using SVM-based classifiers. *Applied Intelligence, 30***(**1), 24–36.<https://doi.org/10.1007/s10489-007-0073-z>

· Islam, M., Haque, M., Iqbal, H., Hasan, M. M., Hasan, M., & Kabir, M. N. (2020). Breast cancer prediction: A comparative study using machine learning techniques. *SN Computer Science, 1***,** 290.<https://doi.org/10.1007/s42979-020-00305-w>

· Sharma, G. N., Dave, R., Sanadya, J., Sharma, P., & Sharma, K. K. (2010). Various types and management of breast cancer: An overview. *Journal of Advanced Pharmaceutical Technology & Research,* ***1***(2), 109–126.

· Jiang, H., Ching, W. K., Cheung, W. S., Hou, W., & Yin, H. (2017). Hadamard kernel SVM with applications for breast cancer outcome predictions. *BMC Systems Biology, 11*(Suppl 7), 138. <https://doi.org/10.1186/s12918-017-0514-1>

· Al-Hadidi, M. R., Alarabeyyat, A., & Alhanahnah, M. (2016). Breast cancer detection using k-nearest neighbor machine learning algorithm. In *2016 9th International Conference on Developments in eSystems Engineering (DeSE)* (pp. 35-39). Liverpool, UK. <https://doi.org/10.1109/DeSE.2016.8>

**APPENDIX:**

**from google.colab import drive**

**drive.mount('/content/drive')**

**Importing libraries**

**import numpy as np**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

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

**from sklearn.preprocessing import StandardScaler**

**from sklearn.svm import SVC**

**from sklearn.neighbors import KNeighborsClassifier**

**from sklearn.linear\_model import LogisticRegression**

**from sklearn.naive\_bayes import GaussianNB**

**from sklearn.model\_selection import GridSearchCV**

**from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report**

**import warnings**

**warnings.filterwarnings('ignore')**

**Data set**

**path = '/content/drive/MyDrive/Colab Notebooks/project/Breast\_cancer\_data.csv'**

**data = pd.read\_csv(path)**

**data.head()**

**data.shape**

**Dropping 'id' and 'unnamed' column**

**data.drop(["id","Unnamed: 32"], inplace=True, axis=1)**

**data.head()**

**data.shape**

**Missing Values**

**data.isnull().sum()**

**data.info()**

**Statistical Description:**

**data.describe()**

**Target [Diagnosis] Variable:**

**malignant\_count = data['diagnosis'].value\_counts().get('M')**

**benign\_count = data['diagnosis'].value\_counts().get('B')**

**print('Number of Malignant cases:', malignant\_count)**

**print('Number of Benign cases:', benign\_count)**

**sns.countplot(data=data, x='diagnosis', palette='Set2')**

**data['diagnosis'] = data['diagnosis'].replace({'M': 1, 'B': 0})**

**data.groupby('diagnosis').mean()**

**Data Copy:**

**df = data.copy()**

**x\_column = df.drop(['diagnosis'], axis=1).columns**

**y\_column = df.columns[0]**

**x\_column**

**Features Distribution w.r.t Diagnosis (M, B)**

**plt.figure(figsize=(8, 10))**

**for i, col in enumerate(data.columns[1:11]):**

**plt.subplot(5, 2, i+1).set\_title(col)**

**x\_col = data[x\_column[i]].values**

**y\_col = data[y\_column].values**

**plt.hist(x\_col[y\_col == 0], label='B', color='green', alpha=0.5, bins=30)**

**plt.hist(x\_col[y\_col == 1], label='M', color='red', alpha=0.5, bins=30)**

**plt.ylabel('Counts')**

**plt.legend(loc='best')**

**plt.tight\_layout()**

**plt.show()**

plt.figure(figsize=(8, 10))

for i, col in enumerate(data.columns[11:21]):

plt.subplot(5, 2, i+1).set\_title(col)

x\_col = data[x\_column[i]].values

y\_col = data[y\_column].values

plt.hist(x\_col[y\_col == 0], label='B', color='green', alpha=0.5, bins=30)

plt.hist(x\_col[y\_col == 1], label='M', color='red', alpha=0.5, bins=30)

plt.ylabel('Counts')

plt.legend(loc='best')

plt.tight\_layout()

plt.show()

plt.figure(figsize=(8, 10))

for i, col in enumerate(data.columns[11:21]):

plt.subplot(5, 2, i+1).set\_title(col)

x\_col = data[x\_column[i]].values

y\_col = data[y\_column].values

plt.hist(x\_col[y\_col == 0], label='B', color='green', alpha=0.5, bins=30)

plt.hist(x\_col[y\_col == 1], label='M', color='red', alpha=0.5, bins=30)

plt.ylabel('Counts')

plt.legend(loc='best')

plt.tight\_layout()

plt.show()

|  |  |
| --- | --- |
| **Separating Features and Target Variable**  X = df.drop('diagnosis', axis=1)  y = df['diagnosis'] |  |

**Visualizing Feature Spread with Swarm Plot**

**def visualize\_spread(X: pd.DataFrame, y: pd.Series, start: int, stop: int):**

**sns.set(style="whitegrid", palette="Set1")**

**data\_standardized = (X - X.mean()) / X.std()**

**data\_combined = pd.concat([y, data\_standardized.iloc[:, start:stop]], axis=1)**

**data\_melted = pd.melt(data\_combined, id\_vars="diagnosis", var\_name="features", value\_name="value")**

**plt.figure(figsize=(10, 10))**

**sns.swarmplot(x="features", y="value", hue="diagnosis", data=data\_melted)**

**plt.xticks(rotation=90)**

**plt.show()**

|  |  |
| --- | --- |
| **Mean features:**  visualize\_spread(X,y,0,10)  Squared Error (SE) features  visualize\_spread(X,y,10,20) |  |

**Correlation Heatmap**

corr\_matrix = X.corr()

fig, ax = plt.subplots(figsize=(22, 10))

sns.heatmap(corr\_matrix, annot=True, linewidths=0.5, fmt=".1f", cmap="Greens", ax=ax)

ax.set\_title("Correlation Matrix Heatmap including all features")

plt.show()

**Feature Selection:**

**drop\_list = ['perimeter\_mean','radius\_mean','compactness\_mean','concave points\_mean','radius\_se','perimeter\_se','compactness\_se','concave points\_se','radius\_worst','perimeter\_worst','compactness\_worst','concavity\_worst','texture\_worst','area\_worst']**

**X\_1 = X.drop(drop\_list, axis = 1 )**

**Correlation Matrix Heatmap after dropping correlated features**

fig, ax = plt.subplots(figsize=(22, 10))

sns.heatmap(X\_1.corr(), annot=True, linewidths=0.5, fmt=".1f", cmap="Greens", ax=ax)

ax.set\_title("Correlation Matrix Heatmap after dropping correlated features")

plt.show()

**Splitting the dataset**

**X\_train, X\_test, y\_train ,y\_test =train\_test\_split(X\_1,y, test\_size=0.2, random\_state=0)**

**X\_train.shape, X\_test.shape, y\_train.shape, y\_test.shape**

**Scaling Data**

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

**Support Vector Machine**

svc= SVC(probability=True)

parameters = {

'gamma': [0.0001, 0.001, 0.01, 0.1],

'C':[0.01, 0.05, 0.5, 0.1, 1,10, 15,20]

}

grid\_search = GridSearchCV(svc, parameters)

grid\_search.fit(X\_train, y\_train)

grid\_search.best\_params\_

svc = SVC(C=20, gamma=0.01, probability=True)

svc.fit(X\_train, y\_train)

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

svc\_acc = accuracy\_score(y\_test, svc.predict(X\_test))

print(svc\_acc)

print(confusion\_matrix(y\_test, y\_pred))

print(classification\_report(y\_test, y\_pred))

cm = confusion\_matrix(y\_test, y\_pred)

plt.figure(figsize=(8, 6))

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', cbar=True)

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.title('SVC model Confusion Matrix')

plt.show()

KNN:

knn = KNeighborsClassifier()

knn\_param\_grid = {

'n\_neighbors': [3, 4, 5, 10],

'weights': ['uniform', 'distance'],

'algorithm': ['auto', 'ball\_tree', 'kd\_tree', 'brute'],

'leaf\_size': [10, 20, 30, 50]

}

knn\_grid\_search = GridSearchCV(knn, knn\_param\_grid)

knn\_grid\_search.fit(X\_train, y\_train)

knn\_grid\_search.best\_params\_

knn = KNeighborsClassifier(

algorithm='auto',

leaf\_size=10,

n\_neighbors=5,

weights='uniform'

)

knn.fit(X\_train, y\_train)

y\_pred\_knn = knn.predict(X\_test)

knn\_acc = accuracy\_score(y\_test, knn.predict(X\_test))

print(knn\_acc)

print(confusion\_matrix(y\_test, y\_pred\_knn))

print(classification\_report(y\_test, y\_pred\_knn))

cm\_knn = confusion\_matrix(y\_test, y\_pred\_knn)

plt.figure(figsize=(8, 6))

sns.heatmap(cm\_knn, annot=True, fmt='d', cmap='Oranges', cbar=True)

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.title('KNN model Confusion Matrix')

plt.show()

log\_r = LogisticRegression()

log\_r.fit(X\_train, y\_train)

y\_pred\_log = log\_r.predict(X\_test)

log\_acc = accuracy\_score(y\_test, log\_r.predict(X\_test))

print(log\_acc)

print(confusion\_matrix(y\_test, y\_pred\_log))

print(classification\_report(y\_test, y\_pred\_log))

**Naive Bayes**

**naive\_b = GaussianNB()**

**naive\_b.fit(X\_train,y\_train)**

**y\_pred\_nb = naive\_b.predict(X\_test)**

**nb\_acc = accuracy\_score(y\_test, naive\_b.predict(X\_test))**

**print(nb\_acc)**

**print(confusion\_matrix(y\_test, y\_pred\_nb))**

**print(classification\_report(y\_test, y\_pred\_nb))**

**Models Accuracy Comparison**

**models = pd.DataFrame({**

**'Model': ['KNN', 'SVM', 'Naive Bayes', 'Logistic Regression'],**

**'Score': [100\*round(knn\_acc,4), 100\*round(svc\_acc,4), 100\*round(nb\_acc,4), 100\*round(log\_acc,4)]**

**})**

**models.sort\_values(by = 'Score', ascending = False)**