BREAST CANCER REPORT

**Introduction**

Breast cancer is one of the most common cancers affecting women worldwide. Early detection and accurate diagnosis are critical for effective treatment and improving survival rates.

This project aims to leverage data science and machine learning techniques to analyze a dataset containing various attributes of breast cancer cases.

The dataset, sourced from the UCI Machine Learning Repository, consists of 574 instances and 32 attributes, including features computed from a digitized image of a fine needle aspirate (FNA) of a breast mass.

**The key objectives of this project are:**

To explore and preprocess the dataset, handling any missing values, duplicates, wrong datatypes, inconsistent data appropriately,.

To engineer new features that could potentially improve the performance of predictive models.

To visualize the data to uncover patterns and insights.

To build and evaluate machine learning models for predicting the diagnosis of breast cancer.

To draw conclusions and provide actionable insights based on the analysis.

**Exploring the Dataset**

**Loading and Inspecting the Dataset:**

The breast cancer dataset was loaded and inspected to understand its structure and content.

The dataset contains 574 rows and 32 columns. Each row represents a different instance of breast cancer, with features derived from a digitized image of a fine needle aspirate (FNA) of a breast mass.

**Identifying Missing Values**

The missing values in the dataset were identified as follows:

|  |  |
| --- | --- |
| Features | Missing values |
| id | 8 |
| Concavity mean | 3 |
| concave points\_mean | 2 |

**Calculating Missing Data Percentage**

The percentage of missing data for each feature was calculated to understand the extent of missing values:

|  |  |
| --- | --- |
| Features | Percentage of missing values |
| id | 1.39% |
| Concavity mean | 0.52% |
| concave points\_mean | 0.35% |

This analysis indicates that the percentage of missing data is relatively low, with all features having less than 2% missing data.

**Handling Missing Data**

**id Column:**

is a unique identifier and missing values in this column do not affect the analysis.

The **id** column serves as a unique identifier and is not required for the analysis or modeling process. Not Dropping rows with missing id values will not impact the quality of our analysis.

**Concavity mean and concave points\_mean columns:**

We will use median imputation to fill in the missing values. This method is simple and effective for numerical data with a low percentage of missing values.

median imputation is a suitable method. It is simple to implement and preserves the overall central tendency of the data.

**Handling inconsistent data**

**Diagnosis column:**

it has categorical data of M and B.

Found two inconsistent values of MM and BB in the diagnosis column.  
   
Chose to use a dictionary to map the inconsistent values to the expected values and using a replace method the values MM and BB were replaced by M and B.

**Handling Duplicate data**Five(5) instances had duplicated data, they had same records with same id with other five rows.

Used a drop method to drop these rows, because this is unnecessary repetition of datasets

**Feature Engineering**

Feature Selection

ANOVA Test  
This applied to assess the significance of continuous features with respect to the categorical target variable. This method identified key features based on their statistical significance (mean, median, mode).

The ANOVA Test was applied to assess the significance of continuous features with respect to the categorical target variable. Higher scores indicate a stronger relationship between the feature and the target

Wrapper Method: Recursive Feature Elimination (RFE)  
Utilized to recursively eliminate the least significant features, helping to identify the most relevant continuous features for the model.

Embedded Method: Lasso Regression  
**Lasso Regression** further refine feature set by shrinking coefficients and eliminating less important features.

Summary of Findings:  
Features like concave points\_worst, perimeter\_worst, area\_mean, Concavity mean, concave points\_mean, radius\_worst, area\_worst, compactness\_mean, concavity\_worst, compactness\_worst, perimeter\_se, area\_se, texture\_mean, texture\_worst were selected.

These features were chosen because they appeared in the results of at least two of the three feature selection methods. This consistency across different methods underscores their relevance and potential impact on the model’s performance.

Reducing Skewness in Data  
To address the skewness in numerical features and improve model performance, the following steps were undertaken:

**Log Transformation:** Applied log transformation to all numerical features. This approach was chosen to reduce the skewness and make the data distribution more normal.

Handling Non-Positive Values During Log Transformation:  
Log transformation is effective for normalizing data distributions and improving model performance. However, applying this transformation to non-positive values (zeros or negative values) can result in invalid or undefined results.

Approach to Handle Non-Positive Values:  
Check for Positive Values  
Handle Non-Positive Values

**Outlier Removal:** Implemented outlier removal using the Interquartile Range (IQR) method. Outliers were identified as data points lying outside 1.5 times the IQR from the quartiles, and these were excluded to reduce their influence on the model.

Impact of Feature Transformation  
The transformation resulted in significant improvements in skewness and kurtosis of the features.

After removing outliers the rows reduced to 507 from 569.

Reduced Skewness: The skewness values are now closer to 0, indicating that the data distribution is less skewed. For most features, the skewness is relatively small, which means that the data is becoming more symmetric.

Improved Kurtosis: The kurtosis values are generally closer to 0, indicating that the tails of the distribution are more in line with a normal distribution. Negative kurtosis values suggest that the data distribution has lighter tails than a normal distribution, which is common after log transformation.

Observations  
The log transformation effectively reduced skewness in most features, bringing them closer to normal distribution.

Outlier removal contributed to a more stable range of feature values and improved the kurtosis of the features.

These transformations enhance the model's ability to learn from data by mitigating the impact of extreme values and non-normal distributions.

Feature Scaling and Normalization  
It ensure that features are on a similar scale which helps many machine learning algorithms perform better.

Standardization (Z-score Normalization):  
Standardization helps in comparing features on the same scale  
Transforms features to have a mean of 0 and a standard deviation of 1.  
Best for algorithms that assume normally distributed data, such as Linear Regression, Logistic Regression, and SVM.

Impact of Standardization:  
**Feature Scales:** All selected features now have a mean of 0 and a standard deviation of 1, which allows for fair comparison across features.

**Feature Values:** The values have been transformed, with the original units of the features being normalized.

Standardizing the features ensures that no single feature dominates due to its scale, which improves the performance and stability of many machine learning algorithms.

The standardized features are now ready for algorithms that are sensitive to the magnitude of features, such as Support Vector Machines (SVM) and Principal Component Analysis (PCA).

Min-Max Scaling (Normalization):  
Scales features to a fixed range, typically [0, 1].  
This method is useful for algorithms that are sensitive to the scale of features and need data to be on a consistent range.Best for algorithms that use distance metrics, like K-Nearest Neighbors and Neural Networks.

Impact of Normalization:  
**Feature Range:** All selected features are now scaled between 0 and 1. This ensures that no feature disproportionately influences the model due to its scale.

**Uniformity:** The scaled data is uniform across features, which can improve the performance of machine learning algorithms that are sensitive to feature magnitudes.

Min-Max Scaling ensures that all features contribute equally to the model’s performance, which can help in achieving more stable and reliable results.

**Data Splitting**

The objective is to evaluate the performance of different machine learning models on standardized data.

Data Splitting with Standardized Data:  
Standardized features have a mean of 0 and a standard deviation of 1 which ensures that each feature contributes equally to the model especially for our data which had no specific range of values but continuous.

Training and evaluation of a model:

Classification Report  
Precision: measures the proportion of true positive predictions out of all positive predictions made by the model.  
High precision indicates that when the model predicts a positive class, it is often correct.

Recall: measures the proportion of true positive predictions out of all actual positive instances in the dataset.  
High recall indicates that the model successfully identifies most of the positive instances.

F1-Score: The F1-Score is the harmonic mean of precision and recall. It balances the two metrics by providing a single score that considers both false positives and false negatives.  
The F1-Score is useful when you need a balance between precision and recall and is particularly valuable when there is an uneven class distribution or when both precision and recall are important.

Support: Support represents the number of actual occurrences of the class in the dataset.  
It provides context on the number of instances that were used to calculate the precision, recall, and F1-Score for each class. High support indicates more data points are available for evaluating the model's performance on that class.

Accuracy: measures the proportion of correctly classified instances (both true positives and true negatives) out of all instances in the dataset.  
High accuracy indicates that the model is performing well overall, correctly predicting the majority of instances.

Macro Average: The Macro Average calculates the average of the precision, recall, and F1-Score for each class, treating each class equally regardless of its frequency in the dataset.  
Macro Average provides a balanced measure of performance across all classes by giving each class equal weight.

Weighted avg: The Weighted Average calculates the average of the precision, recall, and F1-Score for each class, but takes into account the number of instances for each class.  
Weighted Average gives more importance to classes with more instances, reflecting their proportion in the dataset.

Logistic Regression;   
 Achieved the highest accuracy of 97%.  
 It demonstrated excellent precision and recall for both classes.  
 The confusion matrix shows very few misclassifications, indicating the model's robustness in classifying both classes correctly.

Support Vector Machine (SVM);  
 Performed well with an accuracy of 96%.  
 It exhibited strong precision and recall recall.  
 The SVM model had a slightly higher misclassification rate for class 1 (positive cases) compared to Logistic Regression but still provided reliable results overall.

**Cross Validation**

Cross-validation was performed using a 5-fold split to evaluate the performance of various machine learning models on standardized data.

Logistic Regression:  
**Cross-Validation Scores:** [0.9608, 0.9804, 0.9703, 0.9802, 0.9802]  
**Mean Accuracy:** 0.9744  
Logistic Regression demonstrated strong performance with high consistency across folds. The mean accuracy suggests it is a reliable model for this dataset.

Support Vector Machine:  
**Cross-Validation Scores:** [0.9510, 0.9706, 0.9703, 0.9802, 0.9802]  
**Mean Accuracy:** 0.9705  
The Support Vector Machine model yielded robust results with high mean accuracy and minimal variance across folds.  
  
Implications:  
The cross-validation results indicate that models like Logistic Regression, Support Vector Machine, are highly effective for this dataset, with Logistic Regression achieving the highest mean accuracy.

**Recommendations:**   
Given the high accuracy and stability of the Logistic Regression, Support Vector Machine, these models are recommended for further consideration in the analysis.

**Model Evaluation**

Model Training and Prediction:  
Each model is trained on the standardized training data and evaluated on the standardized test data.

**Classification Report:**   
Provides detailed metrics such as precision, recall, F1-score, and support for each class.

**Confusion Matrix:**   
Visual representation of prediction performance, showing true positives, false positives, true negatives, and false negatives.

**ROC Curve and AUC:**   
The ROC curve plots the true positive rate against the false positive rate, while the AUC provides a single metric of the model’s ability to distinguish between classes.

Note:  
True Positives (TP) : The number of instances where the model correctly predicted the positive class.  
False Positives (FP) : The number of instances where the model incorrectly predicted the positive class.  
True Negatives (TN) : The number of instances where the model correctly predicted the negative class.  
False Negatives (FN) : The number of instances where the model incorrectly predicted the negative class.

Logistic Regression  
**True Positives (TP):** 46 (Actual 1, Predicted 1)  
**False Positives (FP):** 1 (Actual 0, Predicted 1)  
**True Negatives (TN):** 103 (Actual 0, Predicted 0)  
**False Negatives (FN):** 3 (Actual 1, Predicted 0)

Conclusion  
Logistic Regression has a high accuracy with 46 true positives and only 3 false negatives, indicating it correctly identifies positive cases well. The model also has very few false positives, showing strong performance in predicting negatives correctly.

Support Vector Machine  
**True Positives (TP):** 46 (Actual 1, Predicted 1)  
**False Positives (FP):** 3 (Actual 0, Predicted 1)  
**True Negatives (TN):** 101 (Actual 0, Predicted 0)  
**False Negatives (FN):** 3 (Actual 1, Predicted 0)

Conclusion  
Support Vector Machine also demonstrates strong performance, with similar true positive and false negative counts as Logistic Regression. It correctly identifies a large number of negatives and has a low number of false positives, indicating effective separation between classes.

**Recommendation  
Both models show high accuracy and good performance metrics. Logistic Regression and Support Vector Machine both have strong capabilities in correctly classifying both positive and negative cases, with minimal errors. The choice between them could depend on specific needs such as model interpret-ability or computational efficiency.**