CLASSIFICATION OF BREAST CANCER USING MACHINE LEARNING MODEL ALGORITHMS.

A dissertation submitted to Manchester Metropolitan University for the degree of Master of Science in the Faculty of Science and Engineering.

By

SOLA PHILIP OYAKALE

This document provides a snippet of the complete project. To request the full document, or to suggest improvements or edits, please contact solaadolf@yahoo.com Additionally, the attached Jupyter Notebook contains further analysis and insights.

PROJECT OVERVIEW

Breast cancer ranks as a foremost contributor to death in women, coming behind lung cancer which causes death in developed, developing and underdeveloped nations of the world. This deadly disease is characterized by symptoms like mutations, frequent pain, breast size and texture changes. Breast cancer is typically classified as benign or malignant; manual classification can be tedious, cumbersome, and inaccurate, resulting in higher error rates.

Machine learning algorithm models like the random forest have demonstrated high accuracy in the classification of breast cancer. In this project, five machine learning algorithm models (logistic regression, K-nearest neighbour, support vector machine, decision tree, and random forest) were used for BC classification. Recent papers on breast cancer classification were reviewed which has made use of supervised machine learning. The major aim of this study was to juxtapose several classifiers and determine the classifier with better accuracy. Logistic Regression achieved the highest accuracy of 96%, outperforming all other classifiers. The accuracy of other classifiers was support vector machine (95%), random forest (94%), K-nearest neighbour (92%), and decision tree (89%).

POTENTIAL PROBLEMS AND SOLUTIONS

The potential problem this project aims to solve is to “classify” breast cancer into either malignant or benign cancer. Classifying cancer helps in knowing the type of cancer, early diagnosis, and treatment of breast cancer. Both machine learning and deep learning often fail to precisely classify images with composite and manifold nature. In the medical image processing domain, breast cancer classification using deep learning presents several challenges. The main challenge is the scarcity of available mammography image datasets, which is a crucial requirement for a deep learning model to achieve better classification accuracy and precision. Therefore, there is a need for larger datasets to improve the training and better understand the classification task (Jabeen et. al. 2023). The feature engineering step often extracts redundant features, leading to a high false-negative rate and long computational time (Jabeen et al. 2023).

AIMS & OBJECTIVES

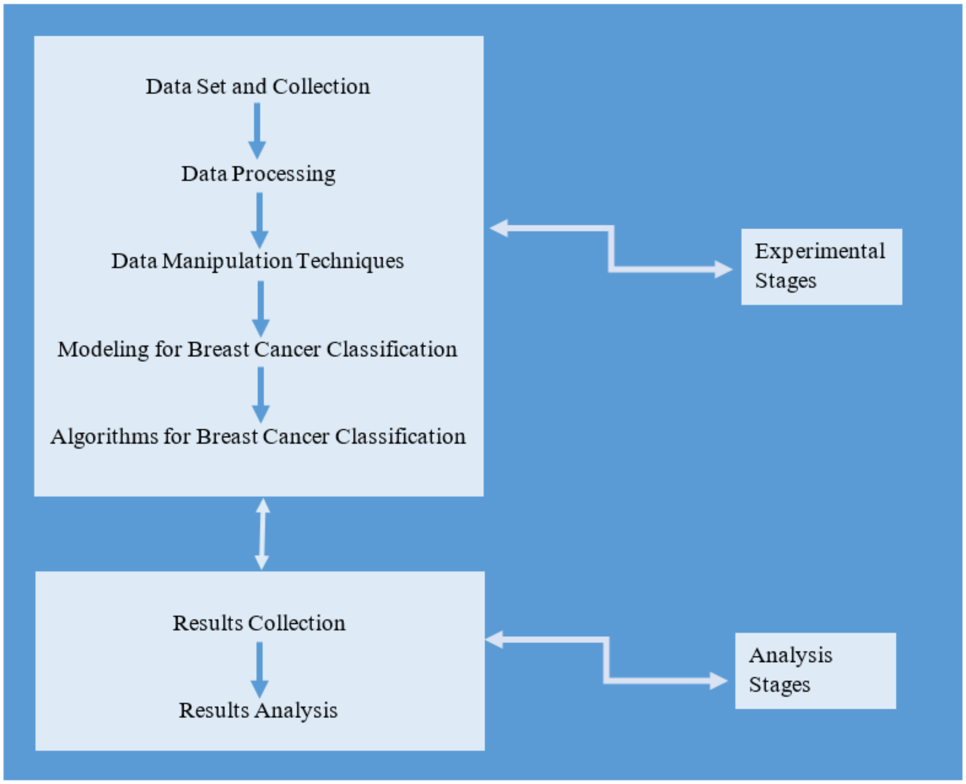
This project aims to explore the use of machine learning algorithms models on our chosen dataset containing variables and other features to classify the type of cancer a patient has either a malignant or Benign tumor.

To accomplish the goals, the following objectives were set.

- To review recent studies for classifying these tumours.
- To classify "breast cancer" types with different machine-learning algorithm models.
- To reduce training time, increase accuracy, and enhance classification.
- To aid in early detection and determine the nature of breast cancer, tumour size analysis is performed.

EXPERIMENTAL METHODOLOGY -OVERVIEW

In data collection & strategy, the source of the data set used techniques in cleaning the data and finding the missing values, visualisation and statistics. Also, strategy is employed to manipulate the data and prepare it for analysis. The next thing is the classification method, i.e. machine learning algorithms predictive models for classification, predictions and analysis, and to evaluate their performances.


Data Set and Collection
Data Processing
Data Manipulation Techniques
Modeling for Breast Cancer Classification
Algorithms for Breast Cancer Classification
Experimental Stages
Results Collection
Results Analysis
Analysis Stages

DATA COLLECTION, IMAGE PROCESSING AND OTHER STRATEGIES

The secondary data set used for this project was downloaded from the UC Irvine machine learning repository you can download via this Link. Breast cancer images can be obtained using various medical imaging techniques such as mammography, ultrasound, MRI, or digital breast tomosynthesis (DBT). Once obtained, the images undergo several preprocessing steps. Firstly, resizing is performed to ensure uniformity in the dataset. Secondly, normalization is done to reduce the effect of variations in image intensity. Thirdly, noise reduction techniques are applied to eliminate any noise that might interfere with the analysis. Lastly, contrast enhancement techniques are used to make features more distinguishable. Techniques like histogram equalization can be useful.

The images are then labelled to indicate whether they are indicative of breast cancer or not. Labels can be binary (cancerous or non-cancerous) or more fine-grained (e.g., benign or malignant). The extracted features and their corresponding labels are then organized into a dataset where each row represents an image, and the columns represent the extracted features and the associated label.

The dataset is then split into training, validation, and testing sets for machine learning model development and evaluation. The extracted features are normalized or standardised, if necessary since some machine learning algorithms perform better with standardized data. Data augmentation techniques can be applied to increase the size and diversity of the dataset. This can be particularly helpful if there are a limited number of images. Lastly, the dataset is stored in a suitable format, such as CSV, or a database, for easy access and analysis.

Overall, breast cancer images can be transformed into structured data and then analyzed using machine learning or deep learning techniques. These analyses can include classification, which means determining whether the image shows cancer or not, segmentation, which involves identifying the boundaries of a tumour, or risk assessment. We will delve deeper into these topics in the next sub-section.

DATA PREPROCESSING AND MANIPULATION

The second step is the preprocessing of the downloaded dataset. The breast cancer dataset was loaded into the machine; unstructured data from the WBCD data set were converted to the structured data set. Following by checking the data head and the info, provided the number of rows and columns while the info gave all the information about the data set i.e. the variables which include but are not limited to id, diagnosis, radius_mean, texture_mean, perimeter_mean, area_mean, smoothness_se, compactness_se, symmetry_se, fractal_dimension_se, etc.

Furthermore, the null value was found with zero returns, this shows that there is no value or no missing value. The column ID was dropped as it is not needed. Another important feature was finding the distribution of the target i.e. diagnosis where Map 'B' to Benign and 'M' to Malignant for the plot labels. Moving to find the features related to the target; finding the correlation to see features that are highly correlated and removing the features that are highly correlated as well. In data analysis and statistics, "highly correlated" refers to a strong relationship or connection that exists between two or more variables.

MODELING BREAST CANCER CLASSIFICATION

In this section, Python (open source programming language; very fast, efficient and easy to learn programming language) and the following libraries (NumPy, pandas, Matplotlib.pyplot, seaborn and sklearn) were used. NumPy simply as "Numerical Python" is a popular Python library for numerical and mathematical operations; especially when dealing with large datasets. Pandas is a popular Python library for data manipulation and analysis, providing data structures and functions. Matplotlib.pyplot is another module within the Matplotlib libraries. Matplotlib was used to create visualisations from scatter plots, charts, and histograms. Seaborn was used to make attractive and statistical graphics easier. It provides a high-level interface for creating aesthetically pleasing and informative statistical plots with less code. Scikit-learn simplifies building and evaluating ML models with tools for classification, regression, and more.

TRAIN & TEST, AND SPLIT

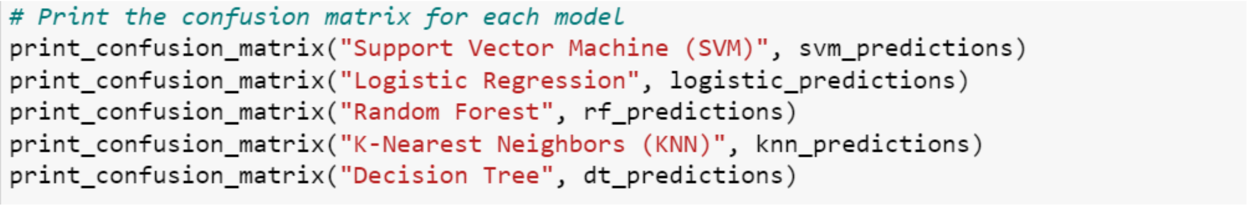
For machine learning to recognize and comprehend patterns, a portion of the actual dataset is provided to it so that the algorithm can learn. As more data is fed into the machine learning algorithm, it becomes more proficient in making decisions based on the patterns it has learned. The model that is generated from the training data is then tested using new and unseen data, to evaluate its performance and determine whether any adjustments need to be made for better results. Essentially, the training data is being used to prepare the model, while the testing data is used to confirm its effectiveness.

The data was splitted into features (X) and the target variable (y). They were further splitted into training and testing sets (x_train, x_test, y_train, y_test = train_test_split). The features were standardised; SVM and KNN benefit from feature scaling. This is important because many machine learning algorithms are sensitive to the scale of the input features. It also helps ensure that all features contribute equally to the model's learning process; scaling also prevents some features from dominating others because of the magnitude differences.

DATA TRANSFORMATION AND MANIPULATION

Furthermore, the five machine learning algorithms (SVM, LR, RF, KNN and DT) used were classified with model, model.fit and predictions. Random_state was set to 42 for SVMs, LR, RF and DT while that of KNN is 5. Evaluating and printing the results for each of the models.

Afterwards, a confusion matrix was employed to evaluate the performance of a classification model. This is important and useful for a model to make sure it is making correct or incorrect predictions. When dealing with imbalanced datasets or different types of errors, confusion matrices are especially useful. The confusion matrix was printed for each model with the code line as displayed below.

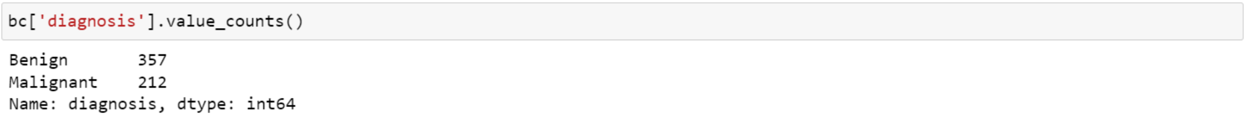


```
# Print the confusion matrix for each model
print_confusion_matrix("Support Vector Machine (SVM)", svm_predictions)
print_confusion_matrix("Logistic Regression", logistic_predictions)
print_confusion_matrix("Random Forest", rf_predictions)
print_confusion_matrix("K-Nearest Neighbors (KNN)", knn_predictions)
print_confusion_matrix("Decision Tree", dt_predictions)
```

To determine the best model, a dictionary was utilized to keep track of the accuracy scores of each model. Model accuracy is a widely used metric to assess the performance of a classification model. This metric measures the proper usage of correct predictions that the model made out of all the predictions it made. When evaluating the performance of a model, it's essential to choose the appropriate metric(s) based on the problem to be solved, the characteristics of the dataset, and the significance of different kinds of errors in our application. Using multiple metrics can often provide a more complete understanding of the model's performance and is therefore good practice to follow. The accuracy_score, was calculated for the 5 models i.e. support vector machine, logistic regression, random forest, K-nearest neighbours, and decision tree. Before proceeding to find the best model based on accuracy.

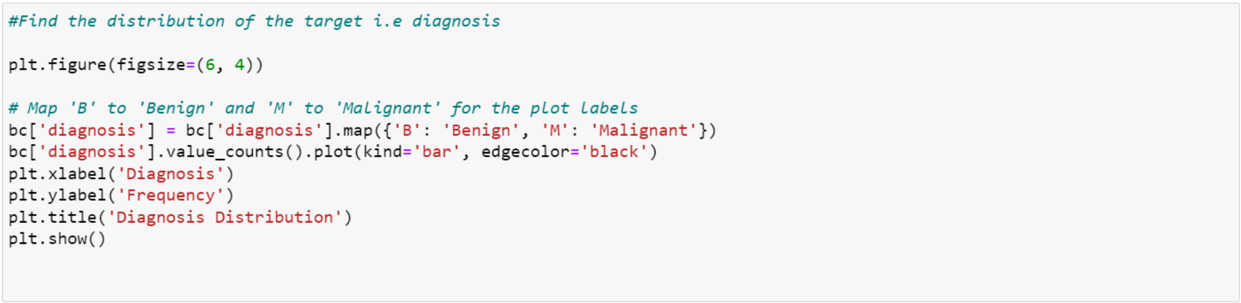
UNIVARIATE ANALYSIS

To understand the characteristics and distribution of the variables, a statistical analytical technique known as Univariate Analysis was utilized. This technique provided valuable insights into the classification tendency, dispersion, and shape of a variable's distribution. It helps to measure the major tendency i.e. mean, median, mode, and measures of dispersion such as variance, standard deviation, and range.



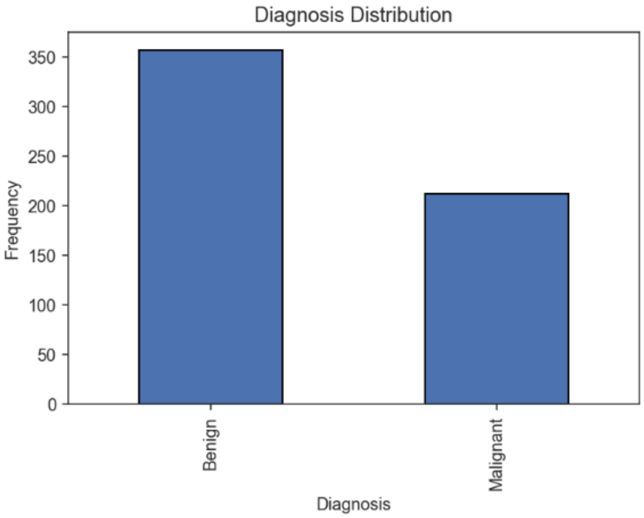
```
bc['diagnosis'].value_counts()
```

```
Benign       357
Malignant    212
Name: diagnosis, dtype: int64
```



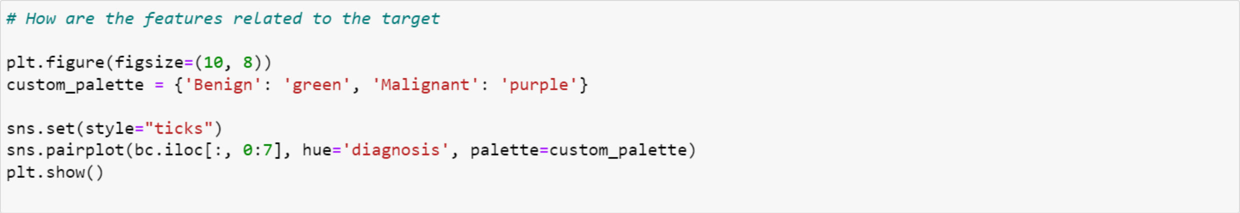
```
#Find the distribution of the target i.e diagnosis

plt.figure(figsize=(6, 4))

# Map 'B' to 'Benign' and 'M' to 'Malignant' for the plot labels
bc['diagnosis'] = bc['diagnosis'].map({'B': 'Benign', 'M': 'Malignant'})
bc['diagnosis'].value_counts().plot(kind='bar', edgecolor='black')
plt.xlabel('Diagnosis')
plt.ylabel('Frequency')
plt.title('Diagnosis Distribution')
plt.show()
```

Additionally, the representation can be in the form of frequency distribution, histograms, bar charts, and box plots for a clearer view. The programming code is given below coupled with the graph representation.



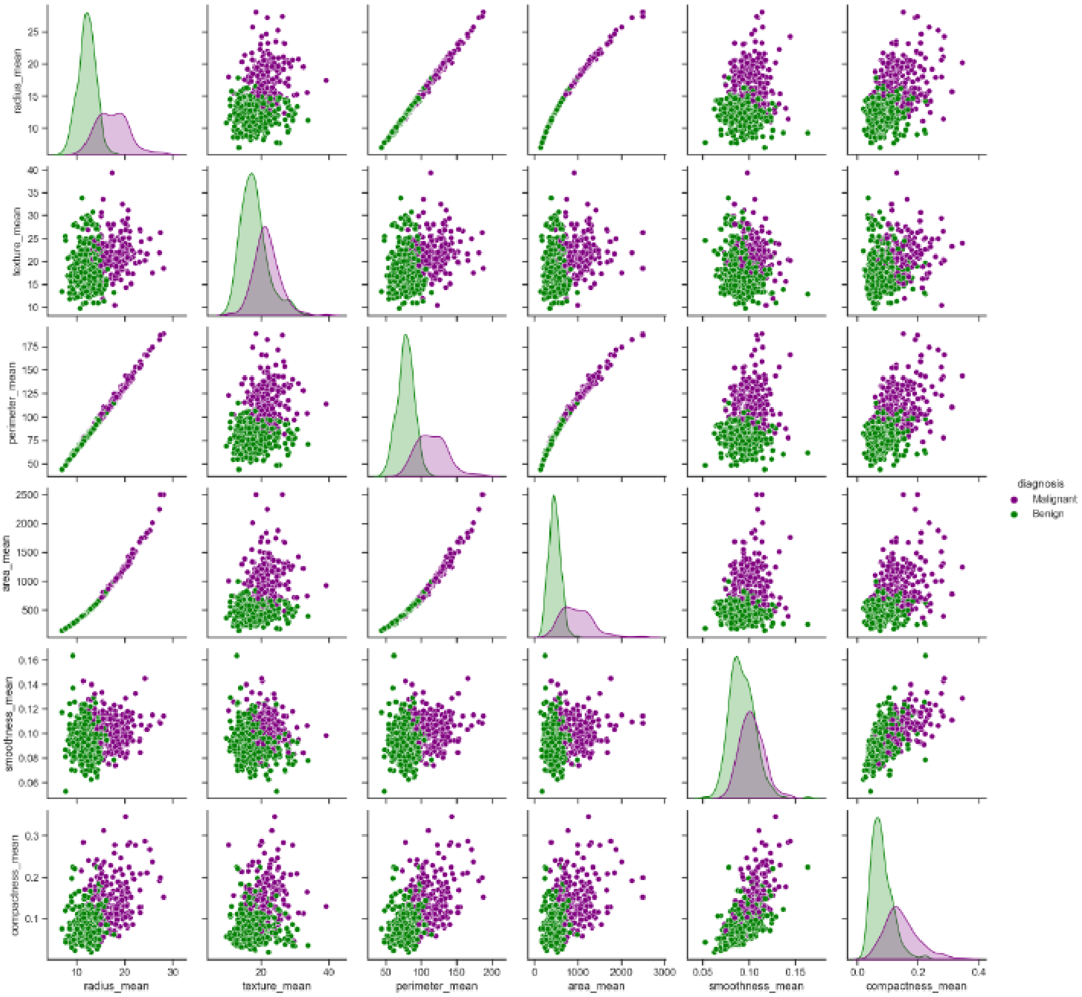
MULTIVARIATE ANALYSIS

In data analysis, there is a statistical technique known as multivariate analysis. This helps to understand the relationship between multiple variables by considering how they interact and collectively influence outcomes or patterns in the data. Some of the common methods used in multivariate analysis are multivariate descriptive statistics, multivariate regression analysis, principal component analysis (PCA), factor analysis, and cluster analysis. Other methods include canonical correlation analysis, discriminant analysis, multivariate analysis of covariance, structural equation analysis, and multidimensional scaling. To understand how the features are represented to the target, we used the programming code below and plotted a graph.

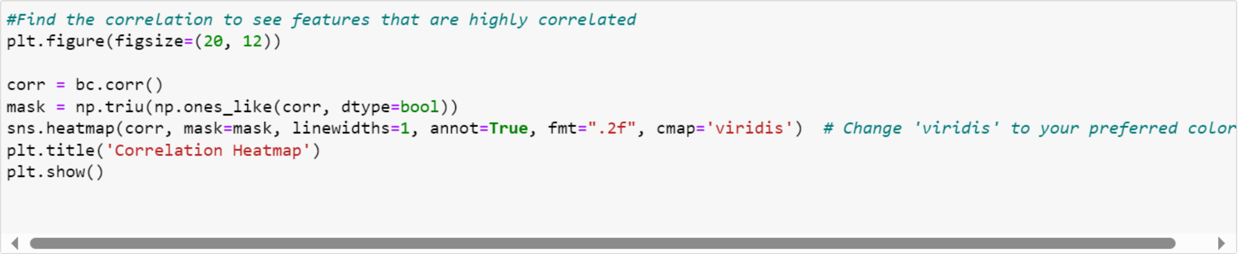


```python
# How are the features related to the target

plt.figure(figsize=(10, 8))
custom_palette = {'Benign': 'green', 'Malignant': 'purple'}

sns.set(style="ticks")
sns.pairplot(bc.iloc[:, 0:7], hue='diagnosis', palette=custom_palette)
plt.show()
```

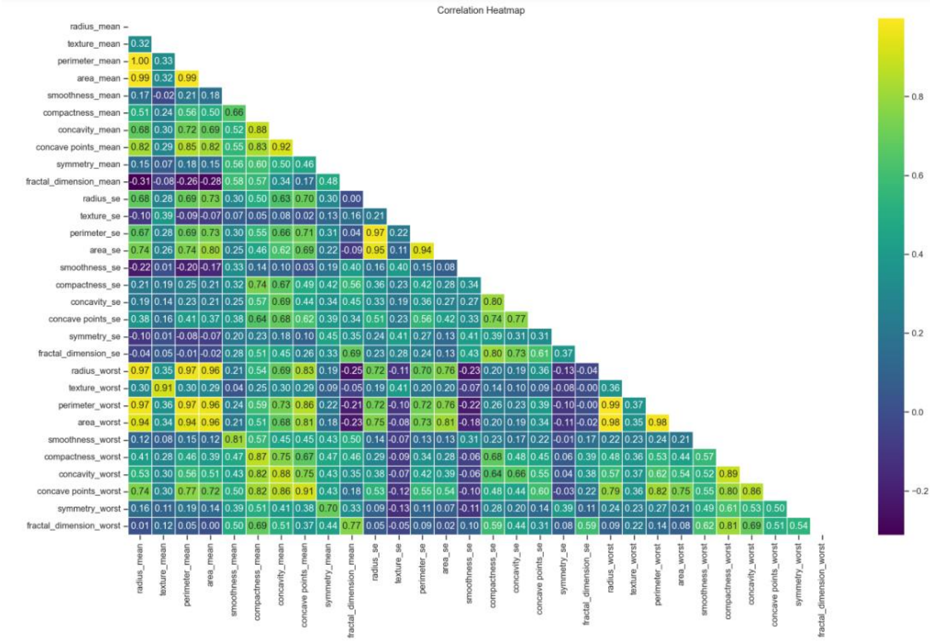
```
<Figure size 720x576 with 0 Axes>
```



The correlation was found to see the features that are highly correlated and were later removed.



```
#Find the correlation to see features that are highly correlated
plt.figure(figsize=(20, 12))

corr = bc.corr()
mask = np.triu(np.ones_like(corr, dtype=bool))
sns.heatmap(corr, mask=mask, linewidths=1, annot=True, fmt=".2f", cmap='viridis')  # Change 'viridis' to your preferred color
plt.title('Correlation Heatmap')
plt.show()
```



Overall, the data was loaded, trained, tested and split. Before evaluating and printing out the results for all the models, the standardised features were classified as well. Other manipulations and techniques are finding the model accuracy, evaluating the performance of a model and many more tasks were performed to have a successful classification and accuracy. The five classification algorithms used in this project are Support Vector Machine, K-Nearest Neighbour, Random Forest, Logistic Regression, and Decision Tree.

K-NEAREST NEIGHBOUR

The K-Nearest Neighbour (K-NN) algorithm is a simple and effective classification rule that is widely used in practice. The k-NN algorithm stores the training patterns and identifies the K-nearest neighbours of a test pattern. It’s capable and can perform both classification and regression.

SUPPORT VECTOR MACHINE

Supervised machine learning algorithms include Support Vector Machines (SVMs) which are utilized for classification and regression tasks. SVMs are particularly useful for classification problems, especially when working with high-dimensional data. They function by segmenting the hyperplane into distinct data.

LOGISTIC REGRESSION

Logistic Regression is mainly used for classification tasks, instead of regression. It is widely accepted due to its simplicity, interpretability, and effectiveness in many applications. Logistic Regression is a simple and fast training algorithm that is easy to interpret. It is a valuable tool for various tasks, and it serves as a solid baseline model for comparison with more complex algorithms.

RANDOM FOREST

Another machine learning algorithm that is highly effective for classification and regression tasks is random forest. Due to its robustness and ability to handle complex datasets, Random Forest is widely used in various applications, including feature selection, regression, and classification. Its versatility and power make it a popular choice for machine learning practitioners.

DECISION TREE

The Decision Tree is a powerful algorithm for supervised machine learning that can handle classification and regression tasks with ease. Each internal node in the tree represents a feature or attribute, and each branch represents a decision rule. The outcome or class label is represented by each leaf node.

APPLICATION PROCEDURES FOR BC CLASSIFICATION

In summary, the process of classifying breast cancer involves using machine learning algorithm models. For better understanding, below is a brief overview of the application procedures used for breast cancer classification:

1. **Data collection:** The dataset was obtained from the Wisconsin Breast Cancer Database (WBCD), which contains features such as tumor size, type, texture, and labels indicating benign or malignant cases of breast cancer.

2. **Data preprocessing:** To work with the data, it needs to be cleaned and preprocessed. This involves handling missing values, scaling features, and encoding categorical variables. After cleaning, the data will be divided into three sets: training, validation, and testing.
3. **Feature selection/extraction:** Informative and important features are selected, such as feature importance analysis and feature engineering, to enhance model performance.
4. **Model selection:** For this project, several machine learning algorithms were selected as models, including SVM, KNN, Random Forest, Logistic Regression, and Decision Tree. Although deep learning models like Convolutional Neural Networks (CNNs) are typically used for image-related research or projects, in this case, machine learning algorithms were utilized for classification purposes.
5. **Model training and evaluation:** Training a machine-learning model involves selecting a dataset and an algorithm to use for training. The quality and size of the training dataset can greatly impact the performance of the model.

   To evaluate the model, relevant metrics such as accuracy, precision, recall, and F1-score are used to assess its performance on a validation dataset.
6. **Model testing:** The performance of the model is tested to make sure there is an unbiased evaluation. This will help us to know or have a proper understanding of each model's performance.

Other important processes that must be considered when developing an application using machine learning. These processes include interpretability and explainability, which involves being able to explain the model's outcomes; deployment, which involves getting the application to the end user; continuous monitoring, which involves ensuring that the application is up-to-date and meeting its intended purpose; documentation and reporting, which involves keeping track of the application's progress and performance; and ethical considerations, which include legal, bias, privacy, and fairness issues that must be taken into account.

EVALUATION METRICS

In the experiments, we used evaluation metrics such as accuracy, precision, recall, and F-measure. The equations used in the calculations include TP (true positives), TN (true negatives), FP (false positives), and FN (false negatives).

RESULTS

Support Vector Machine				
Classification Report	**Precision**	**Recall**	**F1-score**	**Support**
Benign	0.96	0.96	0.96	71
Malignant	0.93	0.93	0.93	43
Accuracy	**Precision**	**Recall**	**F1-score**	**Support**
Macro avg.	0.94	0.94	0.94	114
Weighted avg.	0.95	0.95	0.95	114

Logistic Regression				
Classification Report	**Precision**	**Recall**	**F1-score**	**Support**
Benign	0.97	0.97	0.97	71
Malignant	0.95	0.95	0.95	43
Accuracy	**Precision**	**Recall**	**F1-score**	**Support**
Macro avg.	0.96	0.96	0.96	114
Weighted avg.	0.96	0.96	0.96	114

Random Forest				
Classification Report	**Precision**	**Recall**	**F1-score**	**Support**
Benign	0.96	0.94	0.95	71
Malignant	0.91	0.93	0.92	43
Accuracy	**Precision**	**Recall**	**F1-score**	**Support**
Macro avg.	0.93	0.94	0.93	114
Weighted avg.	0.94	0.94	0.94	114

K-Nearest Neighbors (KNN)				
Classification Report	**Precision**	**Recall**	**F1-score**	**Support**
Benign	0.92	0.96	0.94	71
Malignant	0.93	0.86	0.89	43
Accuracy	**Precision**	**Recall**	**F1-score**	**Support**
Macro avg.	0.92	0.91	0.91	114
Weighted avg.	0.92	0.92	0.92	114

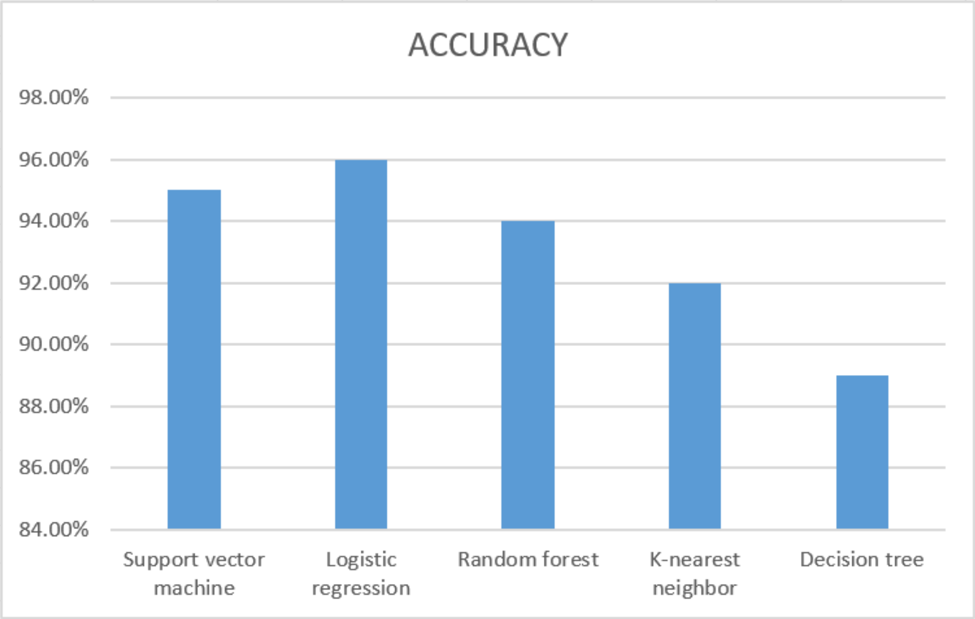
Decision Tree (DT)				
Classification Report	**Precision**	**Recall**	**F1-score**	**Support**
Benign	0.94	0.89	0.91	71
Malignant	0.83	0.91	0.87	43
Accuracy	**Precision**	**Recall**	**F1-score**	**Support**
Macro avg.	0.89	0.90	0.89	114
Weighted avg.	0.90	0.89	0.90	114

DISCUSSION AND COMPARISON

Breast cancer patients always expect accurate results from their examinations with no room for errors. However, there are instances where radiologists provide inappropriate results due to various factors such as the limited number of professionals, large datasets, and inconsistent data values, among others with serious implications that can even result in death. The five machine learning model algorithms used in this project predicted differently with not the same accuracy output and classification. The graphs below show the accuracy of the models followed by the benign and malignant classification.

The table below gives a summary of the models' performance and their accuracy.

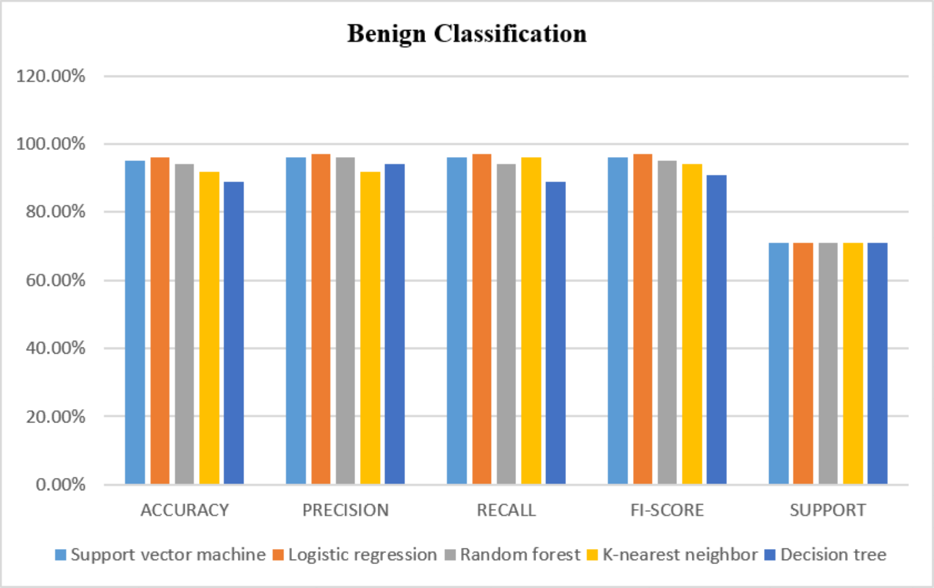
S/N	**MODEL**	**ACCURACY**
1.	Support Vector Machine (SVM)	95%
2.	Logistic Regression	96%
3.	Random Forest	94%
4.	K-Nearest Neighbors (KNN)	92%
5.	Decision Tree	89%



Based on the data classification, the most accurate machine learning algorithm is logistic regression, with 96% accuracy. The support vector machine follows closely behind with a 95% accuracy score. Random forest is in third place with a 94% accuracy, while the K-nearest neighbour algorithm has an accuracy score of 92%. On the other hand, the decision tree algorithm performed poorly, with an accurate score of only 89%.

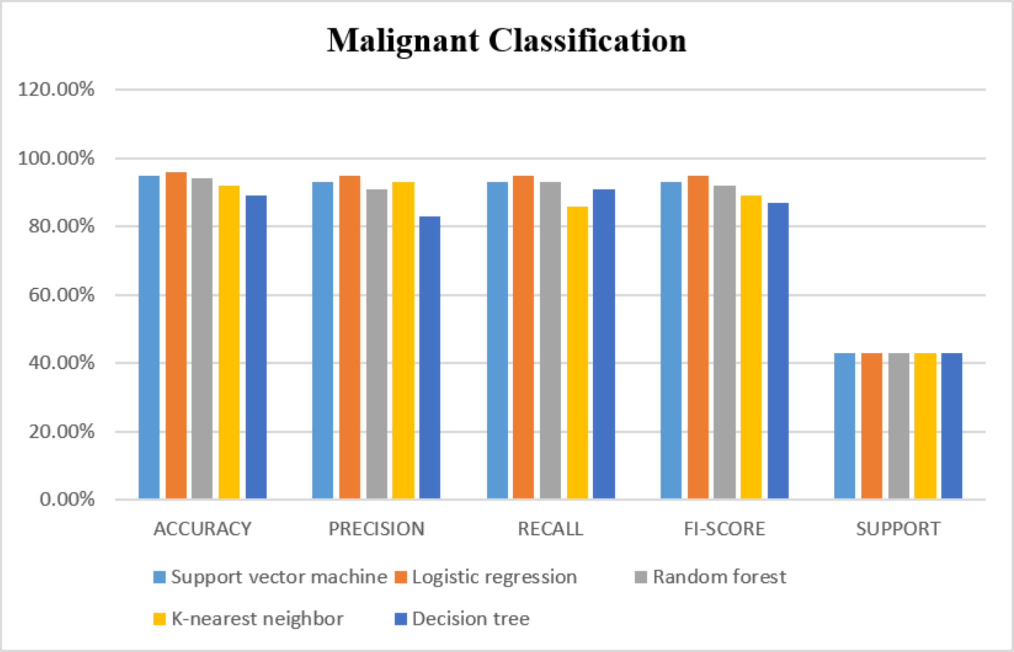
This was further classified into benign and malignant cases separately.

BENIGN: In terms of accuracy parameters, logistic regression and SVMs both scored highest with 96, followed by random forest with 94 and K-nearest neighbor in fourth place with 92. The decision tree came last with a score of 89. For precision, logistic regression performed the best with a score of 97, while SVMs and random forest both scored 96. The k-nearest neighbor had a poor performance with a score of 92, but the decision tree showed an improvement with a score of 94. In terms of recall, logistic regression scored highest with 97, followed by SVMs and random forest with 96. K-nearest neighbor scored 96 as well, while the decision tree came last with a score of 89. As for the F1-score, logistic regression scored the highest with 97, followed by SVMs with 96, random forest with 95, K-nearest neighbor with 94, and decision tree with the lowest score of 91. All the models had the same support value of 71%. The graph below shows the graphical representation of the values.



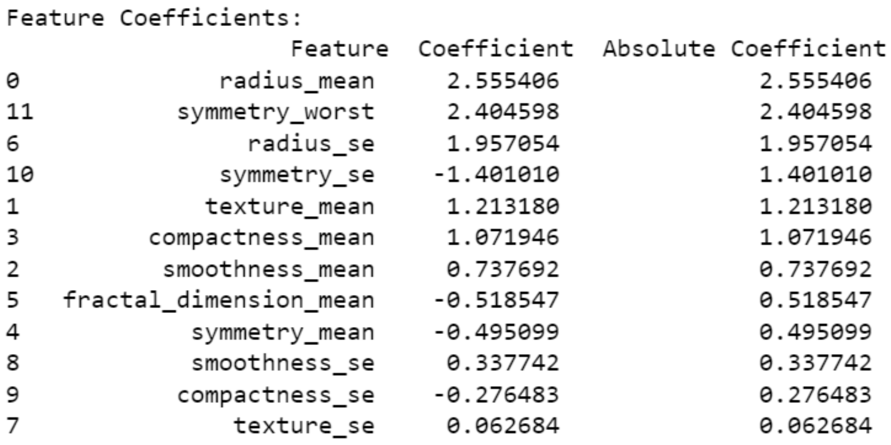
MALIGNANT: All five machine learning models were evaluated based on various parameters such as accuracy, precision, recall, and F1-score. Logistic regression had the highest accuracy of 96%, followed by support vector machine and random forest with 95% and 94%, respectively. K-nearest neighbour and decision tree had accuracy scores of 92% and 89%, respectively. For precision, logistic regression had the highest score of 95%, while the support vector machine and k-nearest neighbour had 93% each. Random forest had a precision score of 91%, and decision tree had the lowest score of 83% for the malignant data. In terms of recall, logistic regression had the highest score of 95%, followed by support vector machine and random forest with 93%. The decision tree showed a significant improvement with 91%, while the k-nearest neighbor had the lowest score of 86%. Logistic regression also had the highest F1-score of 95%, followed by support vector machine and random forest with 93% and 92%, respectively. K-nearest neighbor and decision tree had F1 scores of 89% and 87%, respectively.

All five models had a support score of 43, and they are all represented in the graph below.



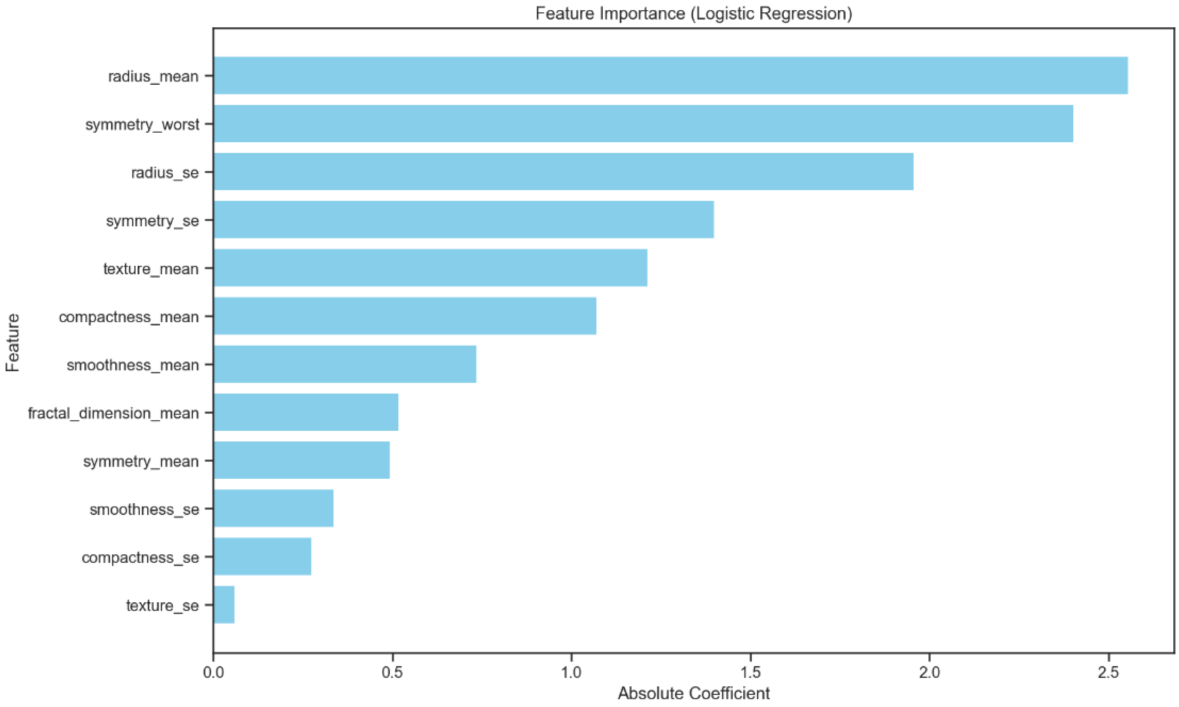
LOGISTIC REGRESSION (BEST CLASSIFIER)

Since the logistic model performed the best of all the machine learning algorithm models. With the regression model, the coefficient (weights) associated with each feature in the logistic regression model were obtained. These coefficients helped to understand the impact of each feature in classifying the data as benign or malignant. DataFrame was created to display each coefficient along with its name and then sorted the DataFrame coefficients by their absolute values in descending order. Finally, the feature coefficients were printed for display, which is displayed below.



```
Feature Coefficients:
                    Feature  Coefficient  Absolute Coefficient
0               radius_mean     2.555406              2.555406
11           symmetry_worst     2.404598              2.404598
6                 radius_se     1.957054              1.957054
10              symmetry_se    -1.401010              1.401010
1              texture_mean     1.213180              1.213180
3          compactness_mean     1.071946              1.071946
2           smoothness_mean     0.737692              0.737692
5    fractal_dimension_mean    -0.518547              0.518547
4             symmetry_mean    -0.495099              0.495099
8             smoothness_se     0.337742              0.337742
9            compactness_se    -0.276483              0.276483
7                texture_se     0.062684              0.062684
```

After completing all the aforementioned steps, which included obtaining the coefficients, creating a DataFrame to exhibit feature coefficients alongside their names, and sorting the DataFrame by absolute coefficient values in descending order, a bar plot visualization was required to indicate the significance of each feature. The y-axis was inverted to display the most important features at the top. As shown below.



CONCLUSION AND FUTURE WORK

The objective of this work is to develop efficient machine-learning model algorithms for classifying breast cancer as either benign or malignant. In this study, a cancer dataset was selected from the UCI Wisconsin Breast Cancer Diagnosis. The main challenge in the field of machine learning is to create accurate classifiers for our analysis. Five algorithms were employed: random forest, support vector machine, random forest, K-nearest neighbor, logistic regression, and decision tree. Key features from our dataset were considered, and feature engineering was done, such as data cleaning, finding null or missing values, and other parameters to ensure robust classification and the best classifier in terms of accuracy.

Logistic regression achieved an accuracy of 96%, surpassing all other classifiers. Support vector machine, random forest, K-nearest neighbor, and decision tree achieved 95%, 94%, 92%, and 89% accuracy respectively. Furthermore, the performance of all five models was evaluated by predicting both malignant and benign cases and analyzing the confusion matrix parameters. The confusion matrix helped to classify the actual benign and malignant cancers. Most papers only use accuracy to evaluate performance, ignoring the confusion matrix.

There are additional areas of classification that can be enhanced in the future or for further investigation, such as improved techniques for feature extraction and determining the performance level of the current model when tested with other breast cancer datasets. Researchers in the future can conduct experiments to determine whether these enhancements can lead to more significant results.

Lastly, it is important to mention that while machine learning was preferred for classification based on the available parameters in the dataset, deep learning was heavily used for breast cancer detection in datasets with images.