Contents

Classification

Introduction

Question 1 examines the classification analysis and development of optimised machine learning models, implementing various data pre-processing techniques, data analysis, feature engineering, feature selection, and dimensionality reduction methods. As a data analyst working on a machine learning project for an enterprise, all methods are required to achieve objectives. The sector assigned with this classification analysis lies in the healthcare services sector, as a result, the dataset was chosen accordingly presenting breast cancer diagnosis information.

Comparison of various supervised and unsupervised machine learning models allowed to choose the three most accurate for breast cancer diagnosis. The choice was based on the problem domain, the advantages, and disadvantages of the models, and their accuracy ranking based on the online research. Information gathered from various journals, articles, scholarly papers, and websites were used to establish the choice of models for classification analysis.

[1]Cleaning, preparing, and manipulating data is not sufficient in the machine learning field of study, before training models and testing data. Therefore, machine learning algorithms require tunning, for this purpose the hyperparameters optimisation was implemented for the Decision Tree and K Nearest Neighbor models.

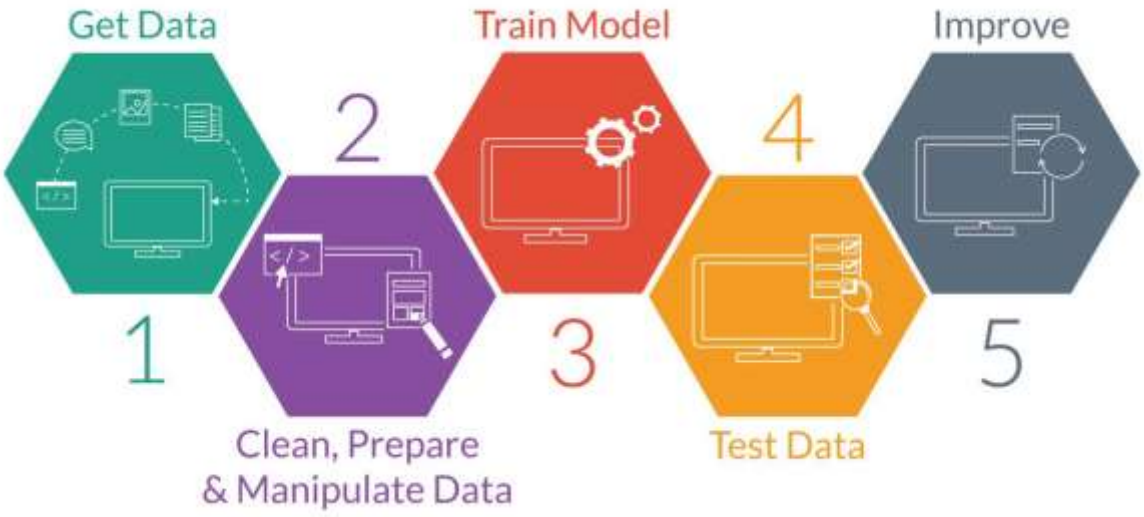


Fig.1.Five Core steps in Machine Learning

Evaluation of all three models aim into establishing the accuracy of all the algorithms including measures and metrics, such as the accuracy score, precision, and recall. Evaluating models allows to discover errors and choose the most accurate one. The comparison of the metrics for this report also considered the objectives of this report, notably for predicting breast cancer diagnosis in the healthcare sector.

Data Cleaning/ Exploratory Data Analysis

The dataset chosen for the analysis is a publicly available CSV file downloaded from the UCI Machine Learning Repository, presenting data on breast cancer examination results of 569 patients.

Before data pre-processing dataset consisted of 33 attributes and 569 records, nevertheless, two variables were removed. One column consisted of null values, whereas another one presented identification numbers of patients. The dataset shape was extracted to 31 variables.

Data pre-processing aims into implementing various techniques to transform the dataset, attaching great importance to removing noisy data and errors. Dataset consists of the real-valued features for each breast cancer cell.

[1] https://cdn-images-1.medium.com/max/1600/1*KzmIUYPmxgEHhXX7SlbP4w.jpeg

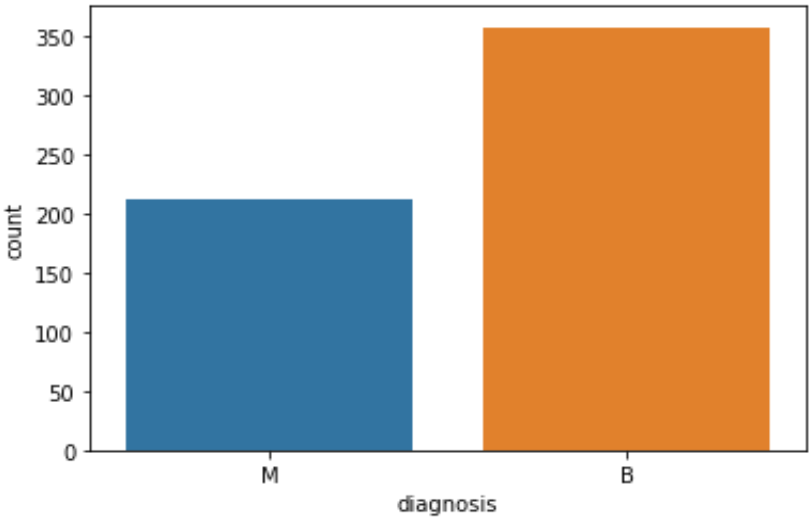


Fig.2. Bar chart of two classifiers: malignant and benign cancer

Figure 1 above presents two classes and the difference between those two. The majority of patients were diagnosed with benign breast cancer compared to the patients with malignant cancer cells.

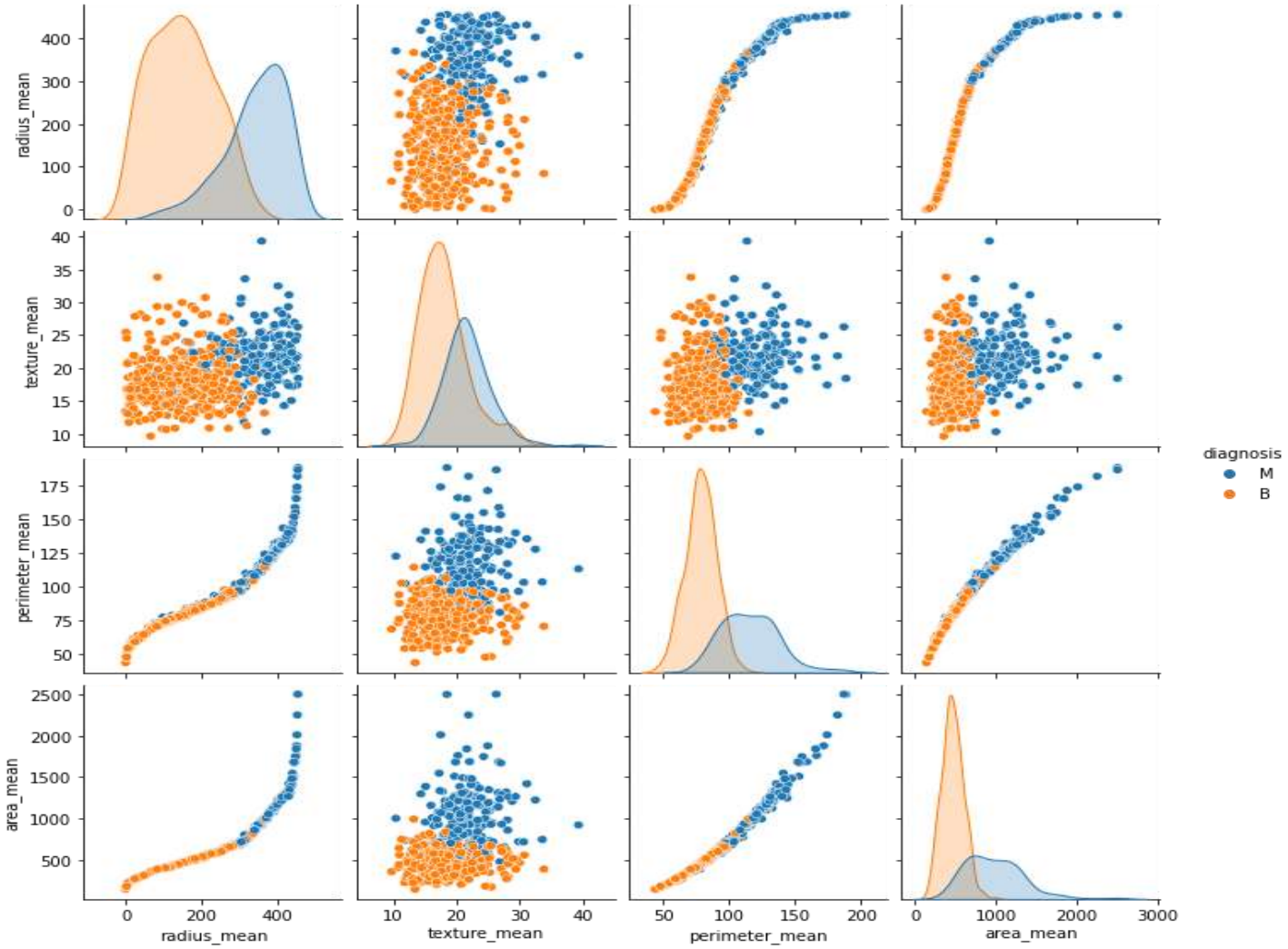


Fig.3. Scatter plot representing patients with malignant cancer in blue colour vs patients with benign cancer as orange colour

	diagnosis
diagnosis	1.000000
radius_mean	0.742827
texture_mean	0.415185
perimeter_mean	0.742636
area_mean	0.708984
smoothness_mean	0.358560
compactness_mean	0.596534
concavity_mean	0.696360
concave points_mean	0.776614

Figure 3 presents the correlation between 9 variables, whereas was established that the strongest correlation occurs between diagnosis variables with radius_mean, perimeter_mean, area_mean,concavity_mean, and concave points_mean attributes.

Fig.4. The strongest correlation between variables

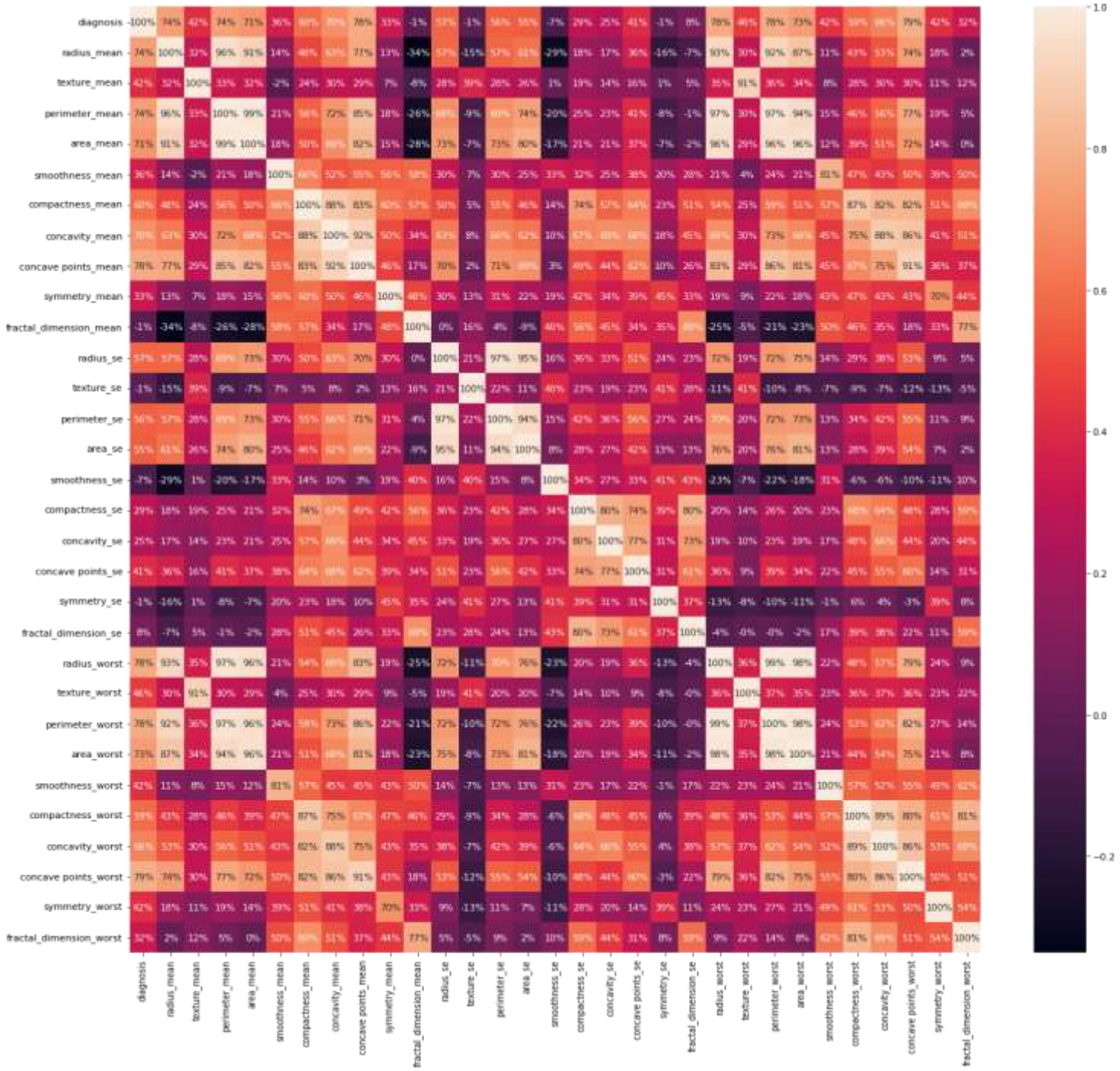


Fig.5. Correlation heatmap on the cleaned full dataset

The brighter the colour of the square, the stronger correlation occurs between the two variables. Visual representation of correlation strength confirms the correlation rates between various attributes with the ‘diagnosis’ variable.

Feature Extraction / Feature Engineering

After implementing data cleaning techniques, feature engineering increases the performance of machine learning algorithms to facilitate the machine learning process. Shekhar (2018) defined feature engineering as *"... the process of using **domain knowledge** of the data to create features that make machine learning algorithms work."* As a data analyst working on a machine learning project for an enterprise, the use of feature extraction is used within statistical analysis techniques, domain knowledge, and maths.

Removing missing values was implemented as a type of feature engineering technique to eliminate noisy data.

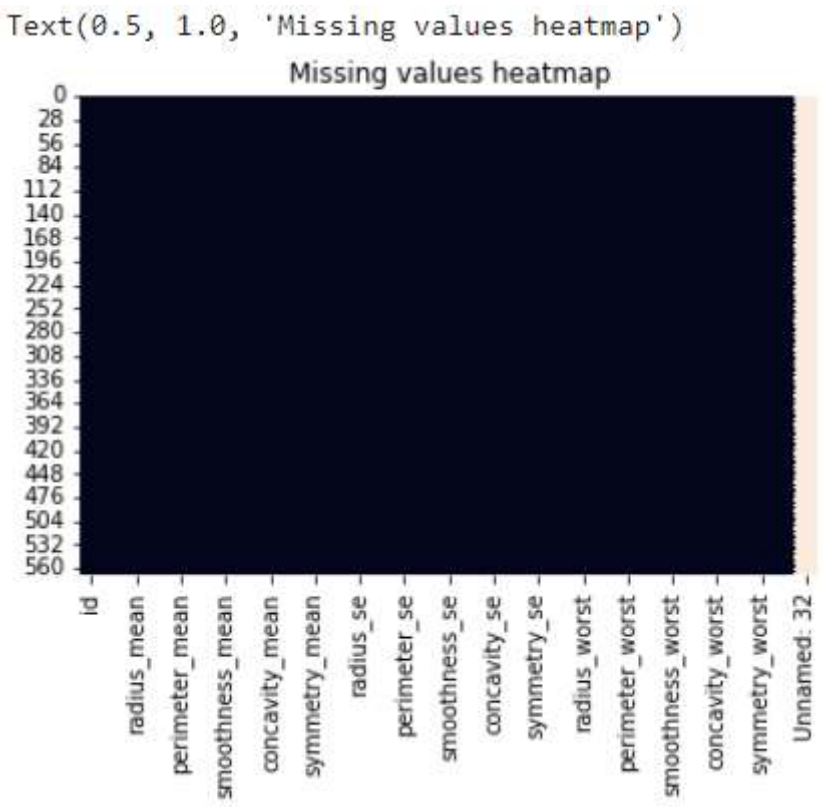


Fig.6. Missing values heatmap - Python

The features created as a result of the classification process allowed for to creation of two classes of 1 for malignant breast cancer cells and 0 for benign breast cancer cells.

M – 1

B - 0

Setting up data for the model by first spitting the dataset into a feature dataset also known as the independent dataset of X plus a target dataset as dependent dataset Y.

```
X = df.iloc[:, 1:31].values
Y = df.iloc[:, 0].values
```

Fig.7. Feature Extraction - splitting data- Python

Furthermore, splitting the data again into 75% training and 25% testing datasets was possible with the use of the '**sklearn'** library. Also, scaling data as the recommended pre-processing method allows for transformation features to avoid any upfront importance. The range set for the feature was chosen to be 0 or 1 as there were two non-numerical values of M and B.

Feature Selection / Dimensionality Reduction

Aiming into reducing the number of input variables the feature selection methods was used for further training of machine learning models. The variable of interest includes information on whether a patient has malignant or benign breast cancer, however additional features of 'radius_mean', 'texture_mean', 'perimeter_mean', and 'area_mean' became included in the subset used for modeling purposes.

	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean
0	M	17.99	10.38	122.80	1001.0
1	M	20.57	17.77	132.90	1326.0
2	M	19.69	21.25	130.00	1203.0
3	M	11.42	20.38	77.58	386.1
4	M	20.29	14.34	135.10	1297.0

Fig.8. Feature Selection of 5 variables

Diagnosis (M – malignant, B – benign)

Splitting data into a training set and a test set is a strategy that uses three parameters of features, target, and test_set size before building models.

Choice of modelling techniques

According to a scholarly paper written in the International Journal of Medical Informatics by several authors, (2020) the most accurate supervised machine learning models for classification used for prediction is the decision tree algorithm. Also, the authors used Support Vector Machine, Naive Bayes, and Decision Forest algorithms, which present high accuracy as well.

Nevertheless, 3 various algorithms were tested with the aim of choosing the most accurate ones for breast cancer detection whether the patient has a malignant or benign type. Presenting the accuracy of all the models together clarifies the choice immediately.

The dependent variable of diagnosis is categorical stating the type of breast cancer diagnosed as a result of the lab examination. Classification machine learning algorithms are used on big data, nevertheless in fact these are used in the classes specifically.

Tree based algorithms

Tree-based algorithms are a common technique used by Data Analysts and Data Scientists according to their high accuracy and ease of interpretation. Decision Tree Classifier can easily capture non-linear patterns and is used for predicting missing values, however unfortunately it can overfit noisy data and cannot be used in big data due to the potential complexity. Bagging algorithms should help with any issues with this model. (www.datacamp.com, 2022).

Decision Tree Classifier is a technique of machine learning for credit scoring and marketing studies, and more importantly, this model is used for establishing a diagnosis of medical conditions based on lab measurements.

Also, this type of machine learning supervised model presents outcomes by a flowchart, which presents decisions as branches and further steps for predicting future success. Figure 8 presents details of two classes of breast cancer diagnosis – Class 0 for **benign, and** Class 1 for **malignant.**

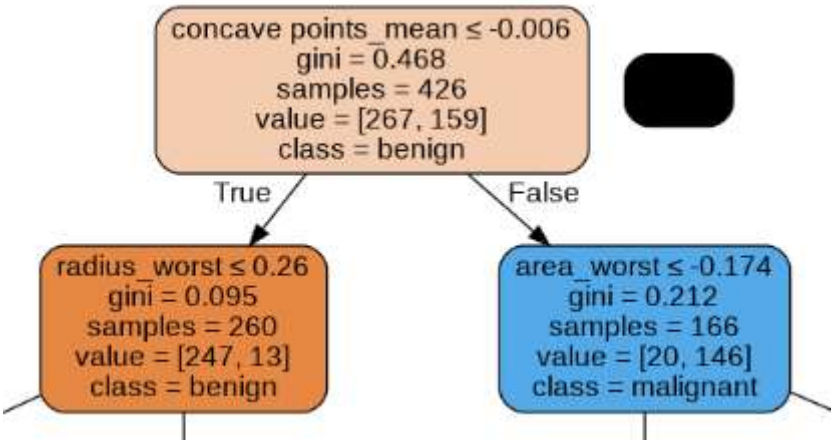


Fig.9. Flowchart obtained by Decision Tree Classifier – Python

Random Forest

[2]As Breiman (2001) defined Random Forest, *"Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest."* Author of the article about random forests aimed into proving why random forests don't overlift as more trees are added. Breiman (2001) explained the rules of the Law of Large Numbers are the reason behind this advantage. *"For the larger data sets, it seems that significantly lower error rates are possible".* It is suggested that different injections of randomness may provide results with more accuracy. In case of the existence of missing values in the dataset, random forest handles these and maintains accuracy as well as handles large data sets with higher dimensionality.

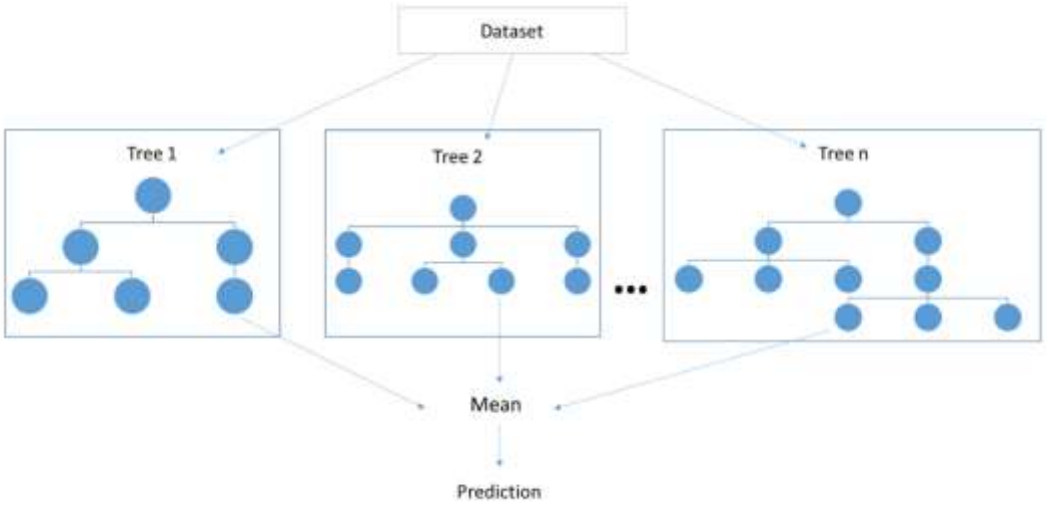


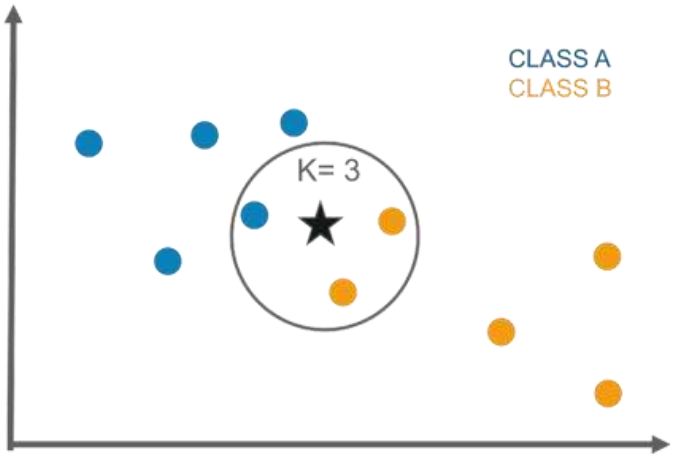
Fig.10. Random Forest algorithm feature selection representation

Despite very little control on a model, the Random Forest modelling technique was chosen for the classification of the breast cancer diagnosis. In fact, the Random Forest model is less prone to overfitting and more accurate than the Decision tree according to Varghese (2018).

K Nearest Neighbor

As Kumar (2020) stated *"KNN is one of the simplest forms of machine learning algorithms mostly used for classification. It classifies the data point on how its neighbor is classified."* KNN is the most common algorithm used in machine learning despite being an unsupervised learning method. This algorithm works to classify new data based on its proximity to the training data.

[2] https://images.deepai.org/glossary-terms/756367e26f1049d3989001c440109fa2/random.png



[3]Fig.11. KNN algorithm visual explanation with k equal to 3

Varghese (2018) stated that the Decision Tree algorithm is more efficient than KNN due to the real-time execution of its performance plus the Decision Tree algorithm supports automatic feature interaction. Figure 11 presents the K = 3 meaning that the 3 closest data are searched to the star data. It is seen that 2 orange circles belong to Class B and 1 to Class A so the star belongs to Class B.

Hyperparameter Optimisation

Yang & Shami (2020) defined hyperparameters as *" ...the parameters that are used to either configure an ML model or to specify the algorithm used to minimize the loss function."* In the aim of fitting a machine learning model into solving a problem domain, it is required to tune its hyper-parameters.

As Prabhu (2018) stated*, "The process of finding the most optimal hyperparameters in machine learning is called hyperparameter optimisation."* Prabhu (2018) used three common algorithms for hyperparameter optimisation, such as Grid Search, Random Search, and Bayesian Optimisation. Therefore, hyperparameter optimisation must be implemented to automate the hyperparameter tuning process for the application of models accordingly and efficiently. (Yang & Shami, 2020).

[3] https://www.edureka.co/blog/k-nearest-neighbors-algorithm/

Decision Trees

Hyperparameters used in modelling the Decision Trees enhance effectiveness across several categories such as controlling the **depth** of the tree, specifying minimum samples before split as well as the use of entropy **criterion**.

Anzar (2020) explained the criterion parameter as *"...the function used to measure the quality of a split and it allows users to choose between 'gini' or 'entropy'."* Gini parameter measures how often the element of the dataset is mislabelled in the process of being randomly labelled. On the other hand, the entropy parameter measures the information indicating the disorder of the features with the target. (Anzar, 2020).

Implementation of three chosen models of Decision Tree, Random Forest, and K Nearest Neighbor included the use of default hyperparameters as the Python libraries were used accordingly. However, controlling the depth of the tree in the Decision Tree flowchart enhances the visibility and transparency of the graph. Also, the 'entropy' criterion was used instead of the Gini index as it is more complex and slightly more accurate.

```
#Using DecisionTreeClassifier
from sklearn.tree import DecisionTreeClassifier
tree = DecisionTreeClassifier(criterion = 'entropy', max_depth=3)
tree.fit(X_train, Y_train)
```

Fig.12. Using the 'entropy' criterion in Python programming when creating a Decision Tree Classifier

K Nearest Neighbor

Essentially, the default value of K is 5 when the Scikit-Learn library is used when programming in Python language is implemented. (Martulandi, 2019). KNN is a non-parametric method, however the hyperparameters of a **number of neighbors**, **distance metric,** and **'p'.** In addition, tuning hyperparameters must be set by a user before implementing machine learning training as it improves model performance.

The number of neighbors is equal to **5**, and the accuracy of the model is equal to 97%.

```
#Using KNeighborsClassifier
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
knn.fit(X_train, Y_train)
```

Fig.13. Setting hyperparameter of number of neighbor to 5 for the KNeighboursClassifier

```
model = models(X_train,Y_train)
```

```
[0]K Nearest Neighbor Training Accuracy: 0.9741784037558685
[1]Decision Tree Classifier Training Accuracy: 0.9929577464788732
[2]Random Forest Classifier Training Accuracy: 0.9976525821596244
```

Fig.14. Presenting models accuracy metric results

If the number of neighbors is equal to **3**, the accuracy is equal to 98%.

```
#Using KNeighborsClassifier
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors = 3, metric = 'minkowski', p = 2)
knn.fit(X_train, Y_train)
```

Fig.15.Setting hyperparameter of number of neighbor to 3 for the KNeighboursClassifier

```
model = models(X_train,Y_train)
```

```
[0]K Nearest Neighbor Training Accuracy: 0.9835680751173709
[1]Decision Tree Classifier Training Accuracy: 0.9765258215962441
[2]Random Forest Classifier Training Accuracy: 0.9976525821596244
```

Fig.16. Presenting models accuracy metric results after changing the number of neighbors

There are 3 distance metrics available, Euclidean, Manhattan, and Minkowski Distances. Within the use of the Scikit-Learn package, the default distance is Euclidean, as a result, the model performance can be improved due to the changing parameters to a different distance. The hyperparameter of the distance of **Minkowski** was chosen for the KNN model used in this classification analysis.

Model Evaluation / Comparison

Evaluation of all three models aim into establishing the accuracy of all the algorithms including measures and metrics. Evaluating models allows one to discover errors and choosing the most accurate one. The comparison of the metrics below also considers the objectives of this report, notably for predicting breast cancer diagnosis in the healthcare sector.

Random Forest Classifier

Metrics below indicate that the best-performed model on the test data is the Random Forest Classifier with an **accuracy** score of 99%.

Accuracy: 98.6%

	precision	recall	f1-score	support
B	0.99	0.99	0.99	90
M	0.98	0.98	0.98	53
accuracy			0.99	143
macro avg	0.99	0.99	0.99	143
weighted avg	0.99	0.99	0.99	143

Fig.17. Random Forest Classifier metrics

Random Forest Classifier leads to 98.6% overall accuracy as a result of the number of correct predictions divided by the total number of predictions. There is 98.6% of observations correctly classified. On the other hand, **precision** indicates the ratio of true positives to the sum of both the true and false positives. The results of 0.99 for B and 0.98 for M confirm that Random Forest Classifier model didn't miss any true positives.

Furthermore, the **recall** measures the sensitivity of the true positives to the sum of both true positives and false negatives. **Recall** measures of 0.99 for B and 0.98 for M.

Random Forest models are formed by a large number of uncorrelated decision trees, which joint together constitute an ensemble. In Random Forest, each decision tree makes its prediction, and the overall model output is selected to be the prediction that appeared most frequently.

K Nearest Neighbor

Accuracy: 95.1%

	precision	recall	f1-score	support
B	0.94	0.99	0.96	90
M	0.98	0.89	0.93	53
accuracy			0.95	143
macro avg	0.96	0.94	0.95	143
weighted avg	0.95	0.95	0.95	143

Fig.18. K Nearest Neighbor metrics

K Nearest Neighbor leads to 95% overall **accuracy** as a result of the number of correct predictions divided by the total number of predictions. There is 95% of observations correctly classified. Another metric of **precision** indicates the ratio of true positives to the sum of both the true and false positives. The **recall** results of 0.99 for B and 0.98 for M.

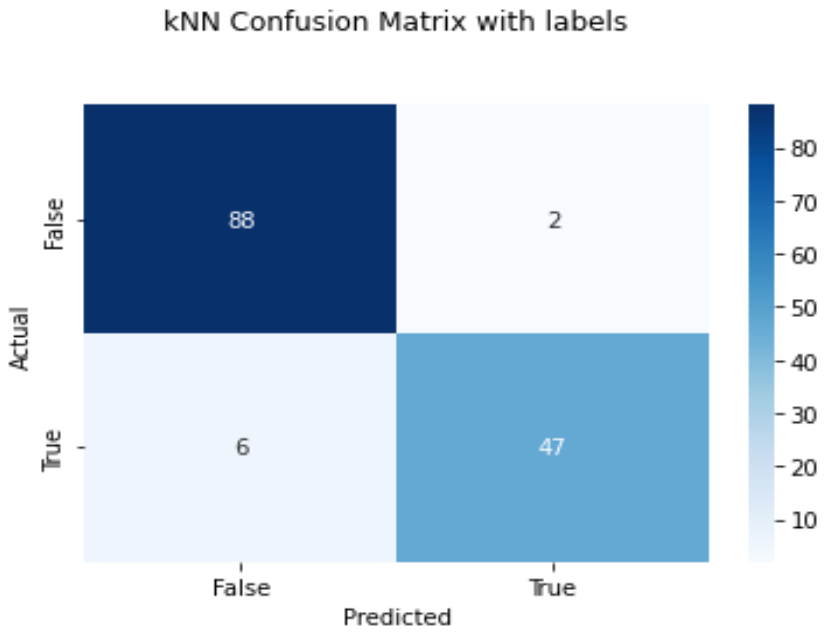


Fig.19. KNN Confusion Matrix

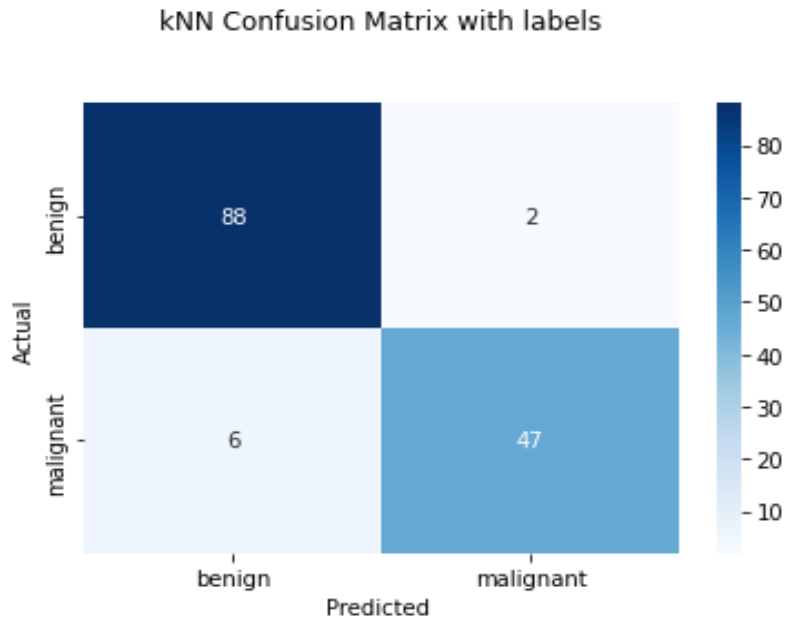


Fig.20.Adjusted KNN Confusion Matrix with benign and malignant labels

"Confusion Matrix is an nxn matrix such that each row represents the true classification of a given piece of data and each column represents the predicted classification." (www.ai-summary.com, 021). Confusion Matrix presents the truth labels versus model predictions as each cell in this matrix represents an evaluation factor. (TP, TN, FP, FN)

M = 1 **B = 0**

Table 1 Confusion Matrix evaluation

True Positive (TP) = 47
The prediction of a malignant type of cancer is true.
True Positive measure of 47 means that those predicted values match the actual values and 47 positive class data points were correctly classified by the model. The prediction that patient X was examined for breast cancer diagnosis has a malignant type of cancer is true.
True Negative (TN) = 88
The prediction of a benign type of cancer is true.
True Negative (TN) measure of 88 means that 8 negative class data points were correctly classified by the model and the KNN model predicted 88 negative values. The prediction is that patient X was examined for breast cancer and diagnosis has a benign type of cancer.
False Positive (FP) = 2
Type 1 Error – The prediction of the benign type of cancer is false.
A false Positive measure means that 2 predicted values do not match the actual values and the KNN model predicted 2 negative values. The prediction that patient X was examined for breast cancer diagnosis has not a benign type of cancer. Prediction of 2 diagnoses of the benign type of cancer is not accurate.
False Negative (FN) = 6
Type 2 Error – The prediction of a malignant type of cancer is false.
False Negative measure means that 6 predicted values do not match the actual values and the KNN model predicted 6 negative values. The prediction that patient X was examined for breast cancer diagnosis has not a malignant type of cancer. Prediction of 6 diagnoses of the malignant type of cancer is accurate. The prediction that patient X was examined for breast cancer diagnosis has a malignant type of cancer whereas was diagnosed with a benign type of cancer.

Overall comment: Prediction is never 100% accurate and it can be seen from the Confusion Matrix that 8 measures in total are false.

Decision Tree Classifier

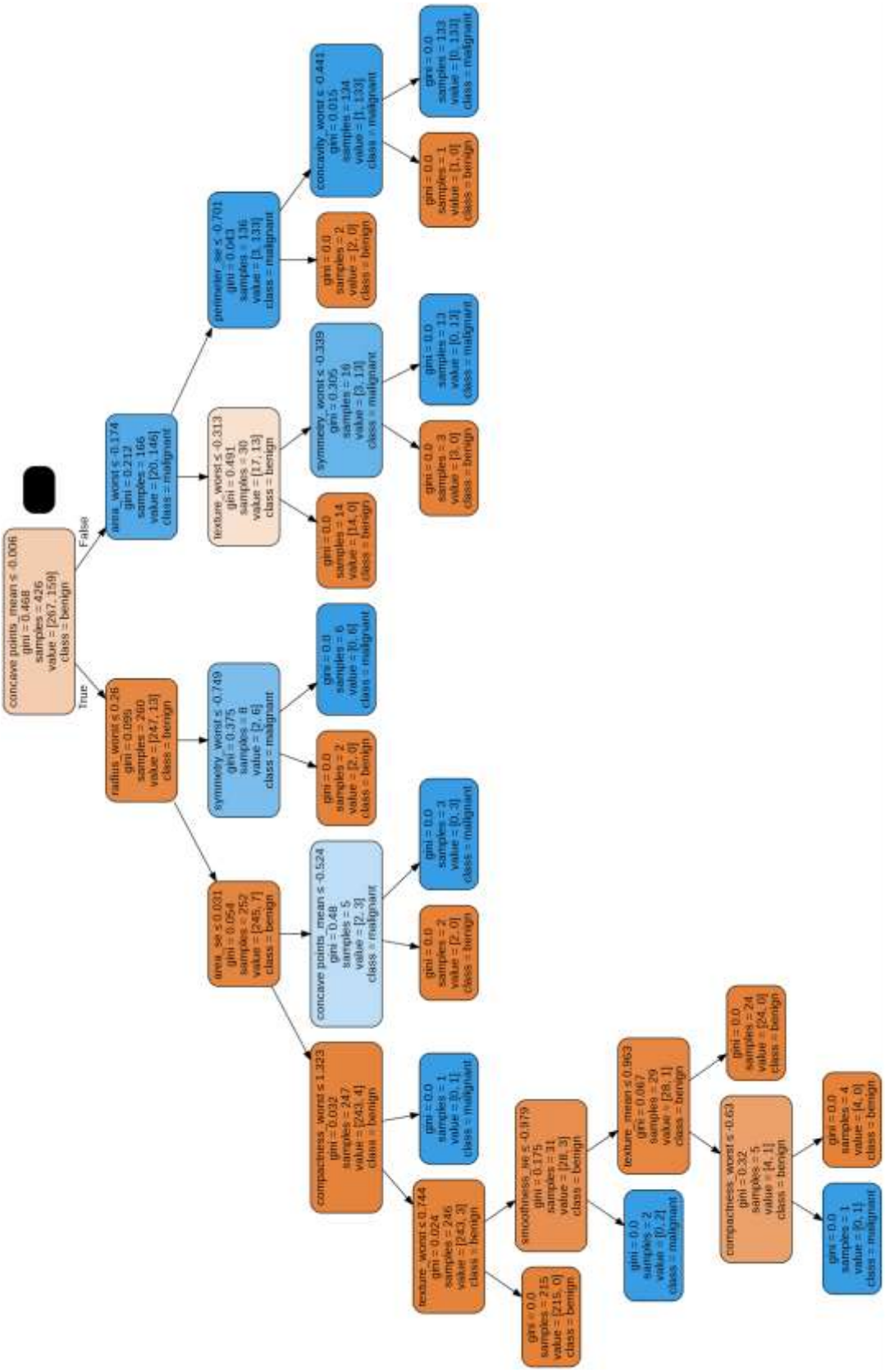


Fig.21. Flowchart of Decision Tree Classifier

Accuracy: 95.1%

	precision	recall	f1-score	support
0	0.99	0.93	0.96	90
1	0.90	0.98	0.94	53
accuracy			0.95	143
macro avg	0.94	0.96	0.95	143
weighted avg	0.95	0.95	0.95	143

Fig.22. K Decision Tree Classifier metrics

Decision Tree Classifier led to 96% overall **accuracy** as a result of the number of correct predictions divided by the total number of predictions. There is 96% of observations correctly classified. Another metric of **precision** indicates the ratio of true positives to the sum of both the true and false positives. The **recall** results of 96% for both B and M confirm that the Tree Decision Classifier model didn't miss any true positives.

Conclusion/Summary

According to WHO breast cancer is prevalent cancer having the 5th leading cause of death. In 2020, 700,000 women died from breast cancer across the world. (www.cancer.net, 2022). *"…there were 7.8 million women alive who were diagnosed with breast cancer in the past 5 years, making it the world's most prevalent cancer."* (www.WHO.int,2021) Early breast cancer diagnosis allows for immediate treatment implementation, which can extend a lifetime.

The use of an optimise machine learning models allows predicting for of information based on big data. The research carried out aimed into establishing the accuracy of the machine learning models in the healthcare sector in predicting cancer diagnosis specifically with the aim of early breast cancer diagnosis to implement treatment immediately.

Shaikh (2022) stated in his paper, *"Manearch teams studied the application of ML and Deep Learning methods in the field of biomedicine and bioinformatics in the classification of people with cancer across high- or low-risk categories."* Essentially, machine learning techniques have features, which allow using various algorithms to train models and predict cancer diagnosis based on the cell information of a patient. Biomedicine and bioinformatics use machine learning algorithms for prediction, classification, and feature selection, whereas various algorithms are helping to increase the quality of treatment by analysing tests and examinations' results. (Kovalenko & Chuprina, 2021).

Evaluation of all three models aimed at the accuracy establishment for all the algorithms including measures and metrics. Evaluating models allows one to discover errors and choosing the most accurate model. The comparison of the metrics above indicated the most accurate model, notably for predicting breast cancer diagnosis in the healthcare sector.

In fact, the machine learning evaluation presents information about the accuracy of the models and this information could be beneficial in the healthcare sector for the data analysts and data scientists for the long-term diagnosis predictions.