Analysis Report

Student Full Name	**Marukh Khan**	**Student ID No.**	**19716844**

Case Study Title
Case Study (A): Predicting Cancer Patients Mortality Status. **Research Question:** Does machine learning have the potential to assist doctors in predicting those who will survive breast cancer or not? **[43 Marks]**

Task (1) – Domain Understanding: Classification

The doctors decided that classification modelling is required. **Indicate in the table below** for each of the listed **variables in your data which ones you should RETAIN** and can be included in the classification modelling of Breast Cancer Mortality (Alive vs. Dead) and **the variables you should DROP (REMOVE). Justify your decision logically and/or by research** (include in-text citation)

[3 Marks]

Variable Name	RETAIN or DROP	Brief justification for retention or dropping with in-text citation.
Patient ID	**DROP**	I dropped this variable as Patient ID is a unique identifier and doesn't contain any meaningful information that is relevant to predicting breast cancer mortality
Month of Birth	**RETAIN**	I kept this variable as month of birth may have an indirect effect on breast cancer mortality, although likely weak.
Age	**RETAIN**	I kept this variable as age is a critical factor in breast cancer mortality, as older patients may have different treatment responses, survival rates and weaker immune systems which would increase mortality risk
Sex	**RETAIN**	I kept this variable as although breast cancer mainly affects females, males can get breast cancer What is breast cancer in men? - NHS Gender influences mortality risk and survival outcomes
Occupation	**DROP**	I dropped this variable as occupation is not a relevant factor is determining whether a patient survives breast cancer
T Stage	**RETAIN**	I kept this variable as T stage refers to the size of tumour, which is important as the size of the tumour can impact survival
N Stage	**RETAIN**	I kept this variable as N stage refers to the cancer spread to surrounding lymph nodes and the increased spread will increase mortality risk.
6th Stage	**RETAIN**	I kept this variable as the Breast Imaging Reporting and Data System (BIRADS) score plays a part in predicting mortality of a cancer patient as a higher score is associated with a worse prognosis, higher maliganancy and a higher mortality rate. The Breast Imaging Reporting and Data System (BI-RADS)
Differentiated	**RETAIN**	I kept this variable as it refers to how the cancer cells look and growing compared with normal cells which play a part in identifying patients who are at a higher risk of mortality
Grade	**RETAIN**	I kept this variable as grade refers to the breast cancer grades which is a critical variable in predicting mortality, as a higher grade of cancer will spread faster, with a

		higher risk of mortality. https://www.macmillan.org.uk/cancer-information-and-suppo and-grading-of-breast-cancer
A Stage	**RETAIN**	I kept this variable as A stage refers to how far the cancer has spread, which is a key predictor in identifying patients at risk as the more spread, the higher the risk of mortality
Tumour Size	**RETAIN**	I kept this variable as tumour size is essential in predicting whether a patient survives or not as larger tumours have a higher risk of spread, which reduces survival chance.
Estrogen Status	**RETAIN**	I kept this variable as cancer cells either have estrogen receptors or not. Research shows that positive hormone receptors are associated with a better chance of survival ER-Positive Breast Cancer: Hormone Receptors, Treatment, and Outlook
Progesterone Status	**RETAIN**	I kept this variable as cancer cells either have estrogen receptors or not. Research shows that positive hormone receptors are associated with a better chance of survival ER-Positive Breast Cancer: Hormone Receptors, Treatment, and Outlook
Regional Node Examined	**RETAIN**	I kept this variable as this is the count of lymph nodes that were examined for cancer spread. The higher number of nodes examined can improve accuracy in identifying type and stages of cancer, which influences treatment decisions, which could increase survival rate.
Regional Node Positive	**RETAIN**	I kept this variable as the amount of regional nodes that are positive can affect cancer mortality risk
Survival Months	**RETAIN**	I kept this variable as it is how long a patient will survive which is directly relevant on predicting mortality
Mortality Status	**RETAIN**	I kept this variable as it is the target variable for our research

References

Healthline, 2025. 'ER-positive breast cancer: Hormone receptors, treatment, and outlook'. Available at: https://www.healthline.com/health/breast-cancer/er-positive-prognosis-life-expectancy [Accessed 01 March 2025]

NHS 2024, 'What is breast cancer in men?' Available at: https://www.nhs.uk/conditions/breast-cancer-in-men/what-is-breast-cancer-in-men/ [Accessed 07/04/2025]

Breast Cancer, 2025 'The Breast Imaging Reporting and Data System (BI-RADS)'. Available at: https://www.breastcancer.org/screening-testing/mammograms/bi-rads-results [Accessed 07/04/2025]

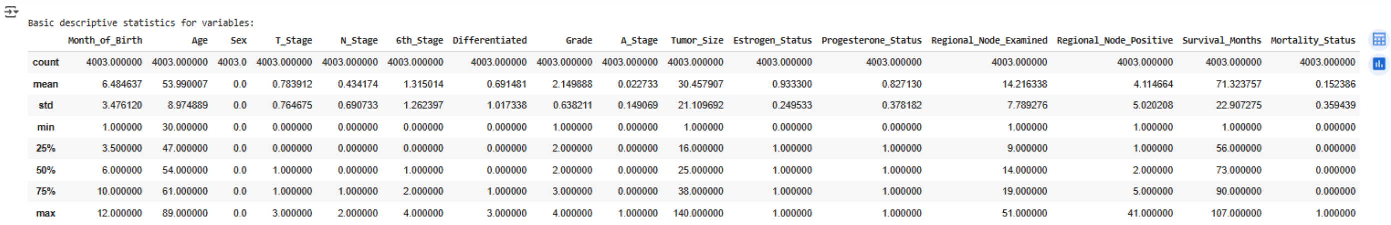
Macmillan cancer support ,2023. 'Staging and grading of breast cancer'. Available at: https://www.macmillan.org.uk/cancer-information-and-support/breast-cancer/staging-and-grading-of-breast-cancer [Accessed 07/04/2025

Task (2) – Exploring and Understanding Your Dataset

With the aid of your Final Python Notebook 1, for your RETAINED input variables and your class "Target" variable, produce 1)**basic descriptive stat**s and 2)**variable scale type, then** 3)p**lot the distribution of your target variable**. (Paste the three screenshots of code OUTPUTS ONLY for evidence of these elements).

[2 Marks]

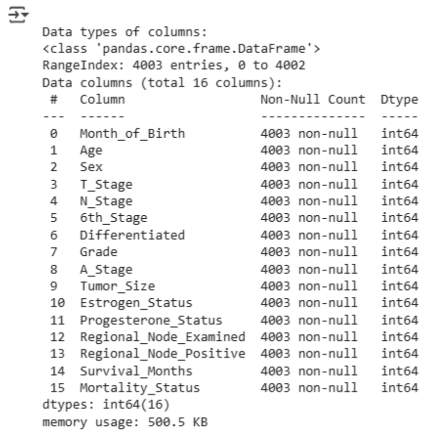
1) Basic descriptive stats



```
Basic descriptive statistics for variables:
```

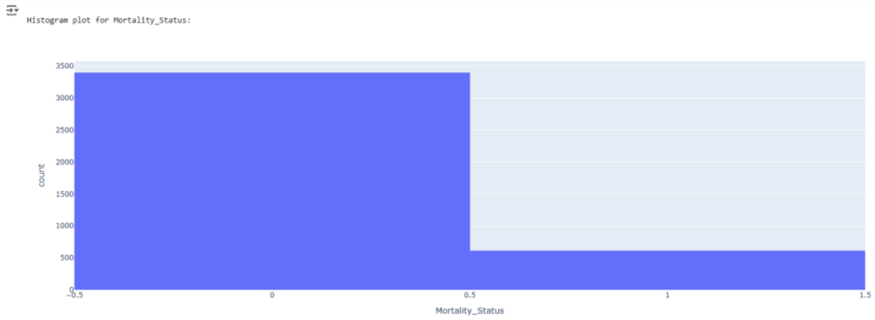
	Month_of_Birth	Age	Sex	T_Stage	N_Stage	6th_Stage	Differentiated	Grade	A_Stage	Tumor_Size	Estrogen_Status	Progesterone_Status	Regional_Node_Examined	Regional_Node_Positive	Survival_Months	Mortality_Status
count	4003.000000	4003.000000	4003.0	4003.000000	4003.000000	4003.000000	4003.000000	4003.000000	4003.000000	4003.000000	4003.000000	4003.000000	4003.000000	4003.000000	4003.000000	4003.000000
mean	6.484637	53.990007	0.0	0.783912	0.434174	1.315014	0.691481	2.149888	0.022733	30.457907	0.933300	0.827130	14.216338	4.114664	71.323757	0.152386
std	3.476120	8.974889	0.0	0.764675	0.690733	1.262397	1.017338	0.638211	0.149069	21.109692	0.249533	0.378182	7.789276	5.020208	22.907275	0.359439
min	1.000000	30.000000	0.0	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	1.000000	0.000000	0.000000	1.000000	1.000000	1.000000	0.000000
25%	3.500000	47.000000	0.0	0.000000	0.000000	0.000000	0.000000	2.000000	0.000000	16.000000	1.000000	1.000000	9.000000	1.000000	56.000000	0.000000
50%	6.000000	54.000000	0.0	1.000000	0.000000	1.000000	0.000000	2.000000	0.000000	25.000000	1.000000	1.000000	14.000000	2.000000	73.000000	0.000000
75%	10.000000	61.000000	0.0	1.000000	1.000000	2.000000	1.000000	3.000000	0.000000	38.000000	1.000000	1.000000	19.000000	5.000000	90.000000	0.000000
max	12.000000	89.000000	0.0	3.000000	2.000000	4.000000	3.000000	4.000000	1.000000	140.000000	1.000000	1.000000	51.000000	41.000000	107.000000	1.000000

2) Variable scale type



```
Data types of columns:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4003 entries, 0 to 4002
Data columns (total 16 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Month_of_Birth          4003 non-null   int64
 1   Age                     4003 non-null   int64
 2   Sex                     4003 non-null   int64
 3   T_Stage                 4003 non-null   int64
 4   N_Stage                 4003 non-null   int64
 5   6th_Stage               4003 non-null   int64
 6   Differentiated          4003 non-null   int64
 7   Grade                   4003 non-null   int64
 8   A_Stage                 4003 non-null   int64
 9   Tumor_Size              4003 non-null   int64
 10  Estrogen_Status         4003 non-null   int64
 11  Progesterone_Status     4003 non-null   int64
 12  Regional_Node_Examined  4003 non-null   int64
 13  Regional_Node_Positive  4003 non-null   int64
 14  Survival_Months         4003 non-null   int64
 15  Mortality_Status        4003 non-null   int64
dtypes: int64(16)
memory usage: 500.5 KB
```

3) Plot the distribution of your target variable, which is mortality status.



```
Histogram plot for Mortality_Status:
```

Task (3) – Data Preparation: Cleaning and Transforming your data

a) With the aid of your Final Python Notebook 1, when you first explored your retained variables in the cancer dataset, you may have found some issues. **1) Report any issues you found in your retained dataset variables**. Based on the issues you found in your data, **2)suggest a suitable possible method to fix each of these issues** and **3)provide your justification for using your suggested fix method**. Use the table below to organise your findings and analysis, and add more rows if needed

[4 Marks]

	Variable Name	Issue found	Proposed fix	Justification for used fix method
1	Occupation	This variable contained 98.93% missing values and is irrelevant in predicting mortality status.	Drop the variable using the drop() function	The variable has extensive missing data and inputting these missing values with the mean would be inaccurate. This variable also doesn't contribute to predicting cancer mortality so retaining it just adds noise to the data.
2	Patient_ID	This variable is irrelevant in predicting mortality status	Drop the variable using the drop() function	Patient ID is a unique identifier which is irrelevant as it doesn’t contribute anything when predicting mortality status and we already have row indices
3	Survival_Months	This variable contains an outlier	Remove the row with the outlier	This extremely high value is likely a data entry error and could distort the model accuracy, so I decided to remove the row
4	Age	This variable contains outliers at indices 142, 212, 522	Remove the rows with the outliers	The ages identified are extreme and are most likely data errors so removing these rows will ensure the model doesn’t contain inaccurate or unrealistic values
5	Regional_Node_Examined	This variable contains many outliers that I identified using the IQR method	Remove the rows with the outliers	High number of lymph nodes examined may not be accurate or it may indicate patient cases with aggressive treatment which does not represent standard patient procedures, so I looked through the list of outliers and removed those I thought necessary

6	Reginol_Node_Positive	This variable is spelt incorrectly	Rename the variable using the rename() function	We rename this variable to ensure consistency
7	Tumor_Size, Age, Regional_Node_Examined, Sex	Missing values	Fill in missing values by replacing them with mean	Inputting the mean into missing values is a standard method for variables when there isn’t a large amount of missing values. It also preserves central tendency and prevents removing data
8	Mortality_Status	Contain values with inconsistent casing	Convert to lowercase using the str.lower() function	This will ensure values are uniform before mapping them to numeric codes
⋮	⋮	⋮	⋮	⋮

Task (3) – Data Preparation: Cleaning and Transforming your data

b) With the aid of Python packages and your Final Python Notebook 1, implement your suggested fixes of issues in the previous Task 3-a in your final Python Notebook1.

1) Show evidence (before and after) of implementing your suggested fix to the problems you identified for your dataset in Task 3-a.

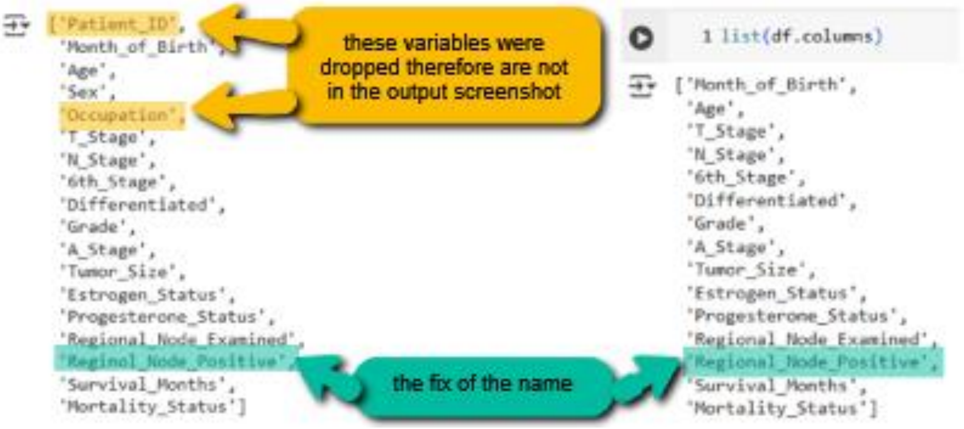
To show your evidence, paste screenshots of your relevant code OUTPUTS ONLY (Do not paste the code).

2) Indicate and annotate the issue and the fix in each of your provided evidence screenshots.

[4 Marks]

Output screenshot of the issue before the fix	**Output screenshot after fixing the issue**

Issues 1, 2 and 6



```
['Patient_ID',
 'Month_of_Birth',
 'Age',
 'Sex',
 'Occupation',
 'T_Stage',
 'N_Stage',
 '6th_Stage',
 'Differentiated',
 'Grade',
 'A_Stage',
 'Tumor_Size',
 'Estrogen_Status',
 'Progesterone_Status',
 'Regional_Node_Examined',
 'Reginol_Node_Positive',
 'Survival_Months',
 'Mortality_Status']
```

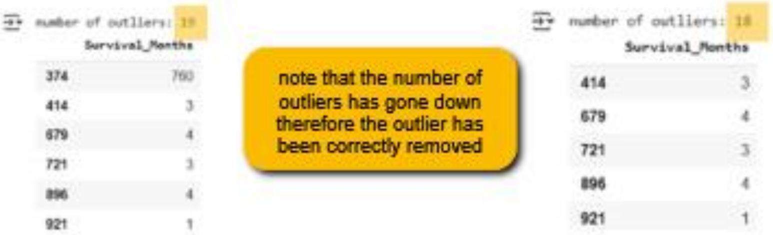
these variables were dropped therefore are not in the output screenshot

the fix of the name

```
1 list(df.columns)
```

```
['Month_of_Birth',
 'Age',
 'T_Stage',
 'N_Stage',
 '6th_Stage',
 'Differentiated',
 'Grade',
 'A_Stage',
 'Tumor_Size',
 'Estrogen_Status',
 'Progesterone_Status',
 'Regional_Node_Examined',
 'Regional_Node_Positive',
 'Survival_Months',
 'Mortality_Status']
```

Issue 3



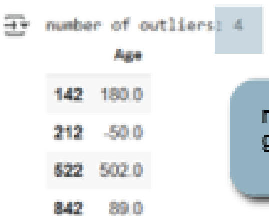
```
number of outliers: 19
     Survival_Months
374              760
414                3
679                4
721                3
896                4
921                1
```

note that the number of outliers has gone down therefore the outlier has been correctly removed

```
number of outliers: 18
     Survival_Months
414                3
679                4
721                3
896                4
921                1
```

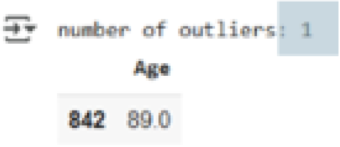
Output screenshot of the issue before the fix	**Output screenshot after fixing the issue**

Issue 4



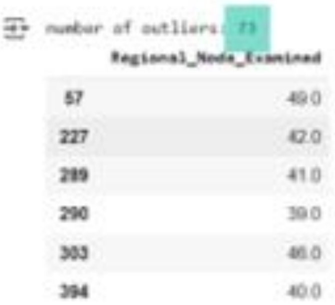
```
number of outliers: 4
        Age
142   180.0
212   -50.0
522   502.0
842    89.0
```

note that the number of outliers has gone down by 3 therefore 3 outliers have been correctly removed



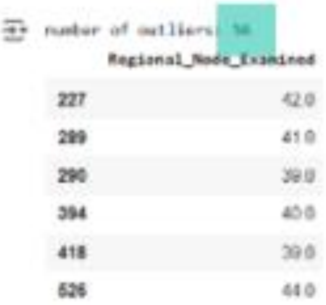
```
number of outliers: 1
      Age
842  89.0
```

Issue 5



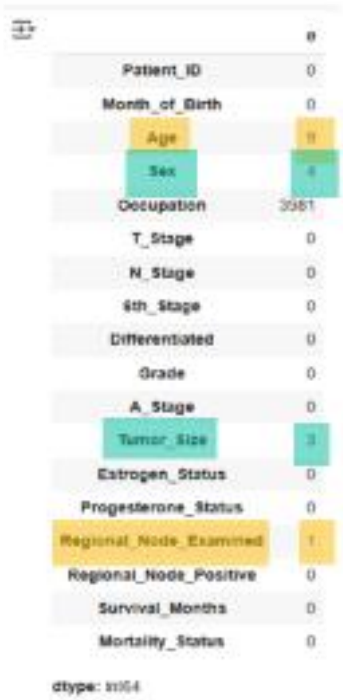
```
number of outliers: 73
     Regional_Node_Examined
57                     49.0
227                    42.0
289                    41.0
290                    39.0
303                    46.0
394                    40.0
```

note that the number of outliers has gone down by 17 therefore 17 outliers have been successfully removed



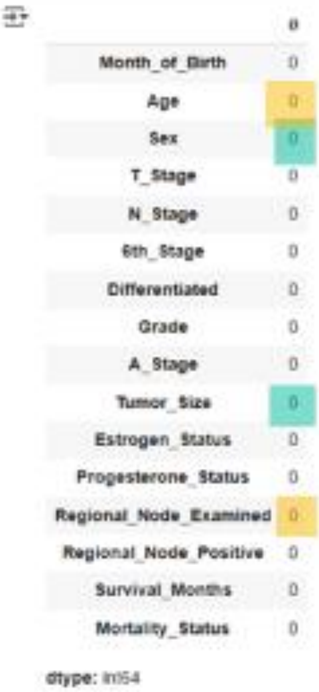
```
number of outliers: 56
     Regional_Node_Examined
227                    42.0
289                    41.0
290                    39.0
394                    40.0
418                    39.0
526                    44.0
```

Issue 7



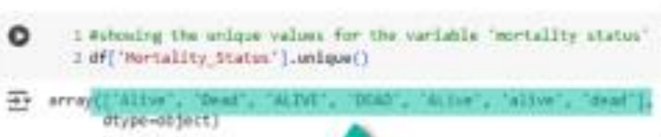
```
                             0
             Patient_ID      0
         Month_of_Birth      0
                    Age      8
                    Sex      4
             Occupation   3981
                T_Stage      0
                N_Stage      0
              6th_Stage      0
         Differentiated      0
                  Grade      0
                A_Stage      0
             Tumor_Size      3
        Estrogen_Status      0
    Progesterone_Status      0
 Regional_Node_Examined      1
 Regional_Node_Positive      0
        Survival_Months      0
       Mortality_Status      0

dtype: int64
```

The issue is the number of missing values and the output sceenshot shows that there are no more missing values



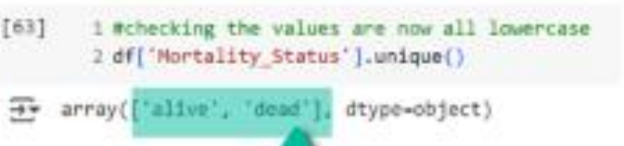
```
                             0
         Month_of_Birth      0
                    Age      0
                    Sex      0
                T_Stage      0
                N_Stage      0
              6th_Stage      0
         Differentiated      0
                  Grade      0
                A_Stage      0
             Tumor_Size      0
        Estrogen_Status      0
    Progesterone_Status      0
 Regional_Node_Examined      0
 Regional_Node_Positive      0
        Survival_Months      0
       Mortality_Status      0

dtype: int64
```

Issue 9



```
#showing the unique values for the variable 'mortality status'
df['Mortality_Status'].unique()
```

```
array(['Alive', 'Dead', 'ALIVE', 'DEAD', 'Alive', 'alive', 'dead'],
      dtype=object)
```



```
[63]
#checking the values are now all lowercase
df['Mortality_Status'].unique()
```

```
array(['alive', 'dead'], dtype=object)
```

this output screenshot before the fix shows inconsistent values with all different casing

here the output values are all consistent ready for mapping

Task (4) – Classification Modelling of Cancer Patients Mortality Status

a) In your Final Python Notebook 2, you built THREE different models to predict cancer mortality status: Logistic Regression (LR), K Nearest Neighbour (KNN) and Naïve Bayes (NB). These algorithms are a mix of parametric and non-parametric algorithms.

1) Note down the type of each algorithm (parametric vs non-parametric),

2) name any learnable parameters, and

3) list any strategic hyperparameters for each algorithm which you want to consider tuning. Organise your answer in the table below:

[3 Marks]

Algorithm Name	Algorithm Type	Learnable Parameters	Some Strategic Hyperparameters
Naïve Bayes (NB)	Parametric as it makes assumptions about the data and estimates parameters (values) from the data	The probabilities of each class	Alpha
Logistic Regression (LR)	Parametric as it assumes a relationship between the input features and the target variable	The coefficients for features	The number of iterations
K-Nearest Neighbour (KNN)	Non parametric as it does not make assumptions	There are no parameters for the algorithm to learn	The number of neighbours (k) to find which value works best for our model. The distance metric used to measure the distance between points (Euclidean, Manhattan)

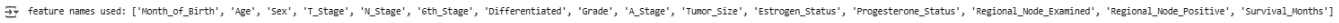
Task (4) – Classification Modelling of Cancer Patients Mortality Status

b) With the aid of your Final Python Notebook 2, use the training–test split approach with your retained applicable input features only and the target output feature to build your predictive classification models.

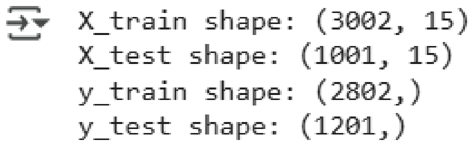
[3 Marks]

i. Screenshot
1) the list of all feature names used for building your classification models and the corresponding
2) data shape function output. (Paste screenshots of the relevant code output only; do not paste the Python code).

1)
2)



```
feature names used: ['Month_of_Birth', 'Age', 'Sex', 'T_Stage', 'N_Stage', '6th_Stage', 'Differentiated', 'Grade', 'A_Stage', 'Tumor_Size', 'Estrogen_Status', 'Progesterone_Status', 'Regional_Node_Examined', 'Regional_Node_Positive', 'Survival_Months']
```



```
X_train shape: (3002, 15)
X_test shape: (1001, 15)
y_train shape: (2802,)
y_test shape: (1201,)
```

ii. In less than 150 words, **research and justify (defend) your choice of the training-test split ratio** and provide an in-text citation.

Splitting data allows you to train a model on one set and then evaluate the model's performance on unseen data. I used a 75:25 training split ratio, with 75% for training and 25% for testing. Using 75% of the data for training allows the model to learn patterns effectively, and the 25% for testing will ensure a reliable evaluation of model performance and prevent overfitting. Selecting an optimal split ratio is crucial to ensure the reliability of the model (Sivakumar et al., 2024)

References

Sivakumar, M., Parthasarathy, S. & Padmapriya, T. (2024) 'Trade-off between training and testing ratio in machine learning for medical image processing', PeerJ Computer Science. Available at: https://doi.org/10.7717/peerj-cs.2245 (Accessed: [26/05/2025]).

iii. Provide as evidence **the code block line** and **code output from** your Final Python Notebook 2 that ensures two conditions:
1) all your models were tested on the same test instances (patients) in your dataset;
2) the labels ratio of Mortality Status "Alive" to "Dead" is the same in the training and test subsets.

3) State the training-test split function parameters in your code line that are responsible for meeting both conditions. **4)** In less than 150 words**, research and justify (defend) your decision to implement both conditions** in your Python Notebook 2 notebook with in-text citations where possible.

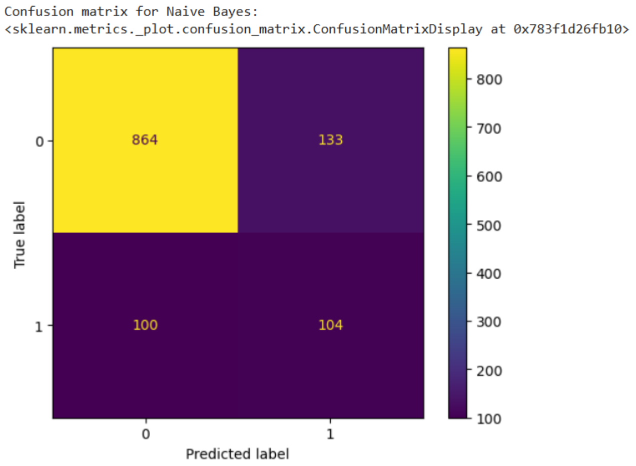
1)
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.25, random_state=42, stratify=y)
 	
2) print("y_train ratio:")
print(y_train.value_counts(normalize=True)) #using value count to count the occurences of each value in the y train
print("y_test ratio:")
print(y_test.value_counts(normalize=True)) ##used to count the occurences of the unique values , not the count
#this shows a percentage of the class labels alive and dead and they should be the same/ similar
 	
3) X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.25, random_state=42, stratify=y)
 	
4)
Testing on the same instances is essential as it ensures fairness and avoids bias in model comparison. It allows direct evaluation of all the models on the same test set, so the differences in performance aren't due to variations in the data, but rather to the model ability. Stratified sampling ensures that the distribution of the target variable (mortality status) is consistent in both training and test sets and improves model fairness. This is why I used stratify=y and set random_state=42 to ensure reproducibility and consistency of results (Scikit-Learn, 2024)

References with in-text citation
Scikit-learn (2024) 'sklearn.model_selection.train_test_split' Available at: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html (Accessed: 02/04/2025]). For q2) Statology (2022) 'Pandas: How to Represent value_counts as Percentage' Available at: https://www.statology.org/pandas-value_counts-percentage/ (Accessed: 09/04/2005)

Task (5) – Evaluating your Cancer Mortality Status Classification Models

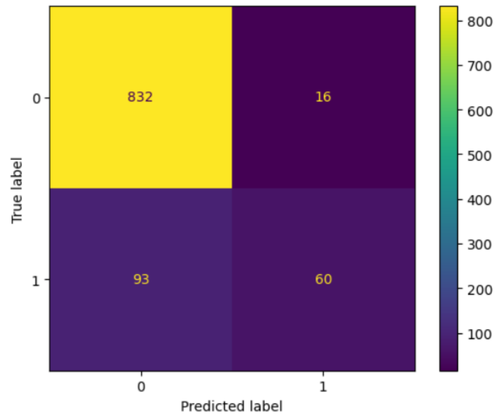
Your healthcare professionals provided the following success criteria to guide you when evaluating and selecting your best model: ***"When evaluating your cancer patients' mortality status classification mode's performance, which addresses your research question. The best model is expected to have some misclassifications. Thus, the model should aim to better discriminate between "Dead" and "Alive" cancer patients"***
a) With the aid of Final Python Notebook 2, for each of your models (Logistic Regression LR, Naive Bayes NB and K-Nearest Neighbours KNN) **1) paste the test confusion matrix,** **2) the classification report and** **3) the AUC-ROC curve graphs** as screenshots from the output of your Python code. **[3 Marks]**

1) Screenshot of the test confusion matrix for (NB, LR, and KNN); make sure you title each matrix with its algorithm name

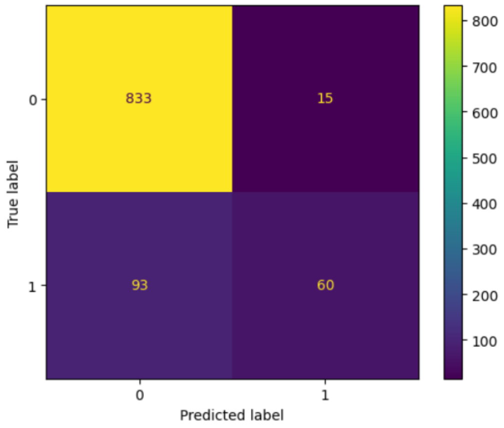


```
Confusion matrix for Naive Bayes:
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x783f1d26fb10>
```

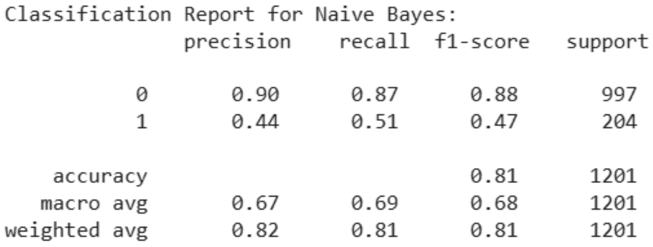
```
Confusion matrix for Logistic Regression
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x783f1fbba790>
```



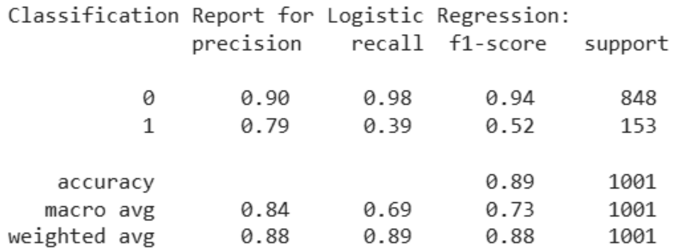
```
Confusion matrix for the best parameters KNN:
```



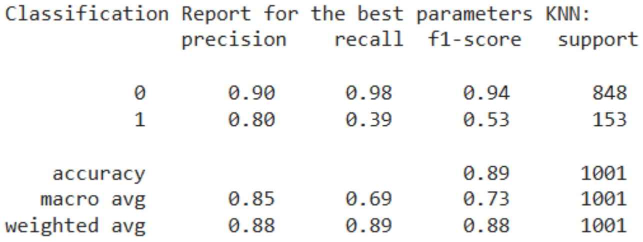
2) Screenshot of the classification report for (NB, LR, and KNN); make sure you title each classification report with its algorithm name



```
Classification Report for Naive Bayes:
              precision    recall  f1-score   support

           0       0.90      0.87      0.88       997
           1       0.44      0.51      0.47       204

    accuracy                           0.81      1201
   macro avg       0.67      0.69      0.68      1201
weighted avg       0.82      0.81      0.81      1201
```



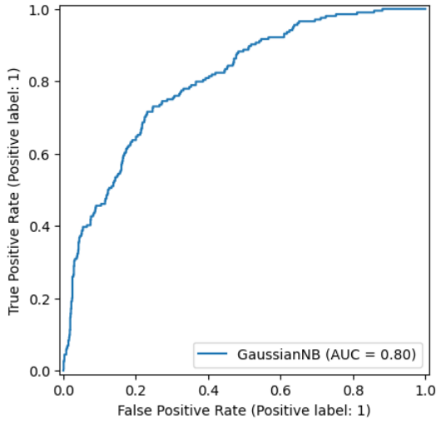
```
Classification Report for Logistic Regression:
              precision    recall  f1-score   support

           0       0.90      0.98      0.94       848
           1       0.79      0.39      0.52       153

    accuracy                           0.89      1001
   macro avg       0.84      0.69      0.73      1001
weighted avg       0.88      0.89      0.88      1001
```



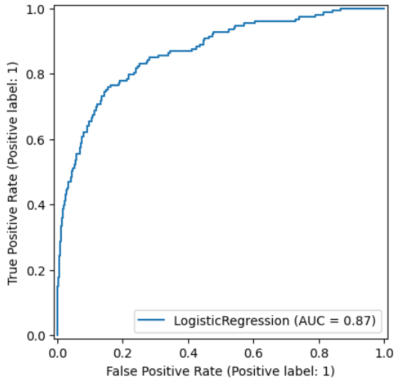
```
Classification Report for the best parameters KNN:
              precision    recall  f1-score   support

           0       0.90      0.98      0.94       848
           1       0.80      0.39      0.53       153

    accuracy                           0.89      1001
   macro avg       0.85      0.69      0.73      1001
weighted avg       0.88      0.89      0.88      1001
```

3) Screenshot of the AUC-ROC Curve for (NB, LR, and KNN); make sure you title each graph with its algorithm name

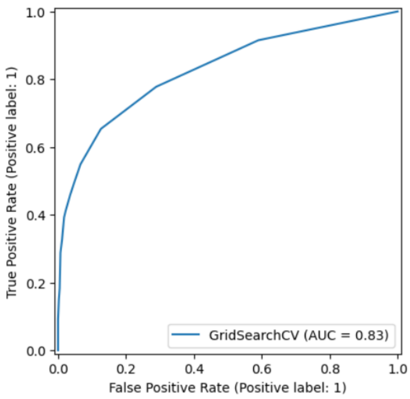
```
ROC Curve for Naive Bayes:
 <sklearn.metrics._plot.roc_curve.RocCurveDisplay object at 0x783f1d0a1450>
```



```
ROC Curve for Logistic Regression:
 <sklearn.metrics._plot.roc_curve.RocCurveDisplay object at 0x783f1d829d10>
```



```
ROC Curve for the best parameters KNN:
 <sklearn.metrics._plot.roc_curve.RocCurveDisplay object at 0x783f1fb8b790>
```



Task (5) – Evaluating your Cancer Mortality Status Classification Models

b) Five different classification evaluation metrics are calculated in your Final Python Notebook 2.

1) State which evaluation metric/metrics to "USE or "DO NOT USE" to closely interpret the above success criteria.

2) For justification, explain how closely your choice of "USE" or "DO NOT USE" for a metric interprets the given success criteria.

With the aid of your Final Python Notebook 2,

3) document all the TEST SCORES for each built model in the table below.

[7 Marks]

Metric	**1) USE or DO NOT USE**	**2) Justification for choosing "USING" or "NOT USING" this metric in relation to the success criteria**	**Model**	**3) Metric Test Score (macro avg)**
Accuracy	USE	Accuracy measures the overall proportion of correct prediction. Higher accuracy indicates strong performance which makes it useful for model selection	NB	0.81
			LR	0.89
			KNN (k=17)	0.89
Recall	DO NOT USE	Recall measures the proportion of actual positives identified correctly but since recall is the same (0.69) for all models, it doesn't help when deciding in which model is best for predicting mortality status	NB	0.69
			LR	0.69
			KNN (k=17)	0.69
Precision	USE	Precision measures the proportion of predicted positive cases that are actually positive, which is crucial when predicting mortality status as we want the model to predict as many patients as possible correctly	NB	0.67
			LR	0.84
			KNN (k=17)	0.85
F1-score	USE	This is an average of precision and recall therefore a higher F1 score indicates overall better model performance	NB	0.68
			LR	0.73
			KNN (k=17)	0.73
AUC-Roc	USE	This measures the models ability to distinguish between classes which makes it crucial for finding the best model.	NB	0.80
			LR	0.87
			KNN (k=17)	0.83

Task (5) – Evaluating your Cancer Mortality Status Classification Models

c) **Suggest a single best mortality status classification model** based on the 'USED' performance metrics scores you identified in Task (5-b). In less than 100 words, briefly **describe how well your best model satisfies the needs of your healthcare professionals.**

[2 Marks]

Based on the performance metric scores, Logistic Regression is the best model for cancer mortality status classification. It achieves the highest AUR ROC and it has high precision and a balanced F1 score.
Since recall is the same (0.69) across all models, it does not help differentiate between them. Therefore, I discarded recall from my decision to find the best model. These metrics indicate that LR is the most reliable model for supporting healthcare professionals in breast cancer mortality prediction.

Task (5) – Evaluating your Cancer Mortality Status Classification Models

d) To enhance your selected best model/s performance (from Task 5-c), tune some of its possible hyperparameters, which you indicated in Task (4-a) for that specific algorithm. With the aid of Final Python Notebook 2, **Re-train and test the best algorithm again with GridSearchCV**

[5 Marks]

i. With the aid of your Final Python Notebook 2**,**
1) Paste into this report the line of code which shows evidence of specifying a parameters grid and applying the GridSearchCV function to rebuild your selected best model.
2) Then, document the estimated best hyperparameters for the optimised model.

1)

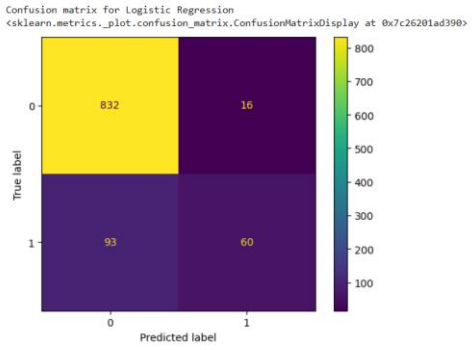
```
# defining the parameter grid for Logistic Regression
param_grid = {
    'C': [0.001, 0.01, 0.1, 1, 10, 100, 1000],
    'max_iter': [100, 1000, 2500, 5000],   # the max no of interations for optimisation
    'solver': ['liblinear'],
    'penalty': ['l1', 'l2']  # penalty controls the type of regularisation and can help for better tuning
}
# applying the GridSearchCV and fitting it
optimised_lr = GridSearchCV(lr, param_grid, cv=5)
optimised_lr.fit(X_train, y_train)
```

2)

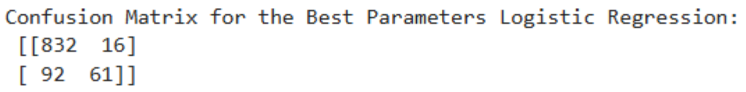
```
{'C': 1, 'max_iter': 1000, 'penalty': 'l1', 'solver': 'liblinear'}
```

ii. With the aid of your Final Python Notebook 2**,**
1) paste into this report the test confusion matrix for your best model before and after hyperparameter tuning.
2) Also, document the new score/s of the "USED" performance metric/s of your choice to interpret the success criteria indicated in Task (5.b) before and after tuning.
3) Comment on whether the tuning of hyperparameters of your best model improved its positive predictive ability in line with the success criteria.

1)



```
Confusion matrix for Logistic Regression
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7c26201ad390>
```



```
Confusion Matrix for the Best Parameters Logistic Regression:
 [[832  16]
 [ 92  61]]
```

2)
After tuning the parameters to optimise my Logistic Regression model, the performance metrics taken from the classification report mostly stayed the same. Both accuracy, precision, F1 score and AUR-Roc curve stayed the same, which shows that the hyperparameter tuning did not lead to a significant improvement in the model's performance.

3)

Before tuning, the model had a precision of 0.79, which meant it was able to detect many positive cases. After tuning, the precision stayed the same at 0.79. The F1 score for class 1 slightly increased from 0.52 to 0.53 showing a minor improvement in the average of precision and recall. Although there is a slight improvement, the model is still not better at identifying positive cases.

Task (5) – Evaluating your Cancer Mortality Status Classification Models

e) Based on your selected best model, **1) criticise your best-performing model**, and **2) state any limitations you may have identified** and **3) any ethical issues your model may raise** if used for predicting breast cancer mortality status.

[2 Marks]

1) Logistic regression has some weaknesses that could impact performance. It assumes that there is a linear relationship between the input features and the target variable/ outcome, whereas, in reality, cancer and the relationship between features and mortality are complex. This model is also simple and doesn't capture complex interactions.

2) Bias is a situation where the models predictions are in favour of a certain outcome due to biased or unrepresentative data. This could be sampling bias, historical bias where the data is from the past, bias in medical data or feature bias.

3) When creating a model in machine learning for healthcare, patients must be informed about how their data will be used, and if not correctly informed it could be unethical.
This model might contain bias, which can cause it to learn and make inaccurate predictions for different groups resulting in a discriminatory and unfair outcome.
Transparency and understanding may cause an ethical issue as healthcare professionals will need to understand how the model makes predictions and relay this to the patients as it is about their breast cancer mortality status prediction. If the model is too complex, it might not be clear as to how it arrived at a decision. This could also affect doctors trust in the model.

Task (5) – Evaluating your Cancer Mortality Status Classification Models

f) With the aid of your Final Python Notebooks 3, combine only TWO out of the THREE base learners (NB, LR, KNN) that you already built into a probability-based voting ensemble classifier.

[5 Marks]

i. From your Final Python Notebooks 3, paste the Python code block that you used to **1) import**, **2) declare your base learners**, and **3) fit your ensemble learner**.

1)

```
knn = KNeighborsClassifier()
nb = GaussianNB()
```

2)

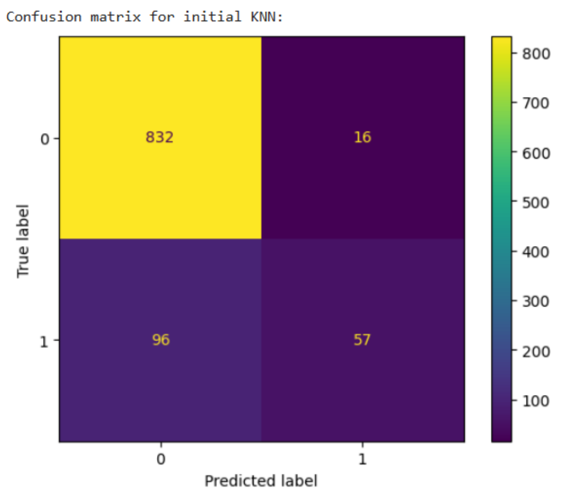
```
base_learners = [('knn', knn), ('nb', nb)]
ensemble_learner = VotingClassifier(base_learners, voting='soft')
```

3)

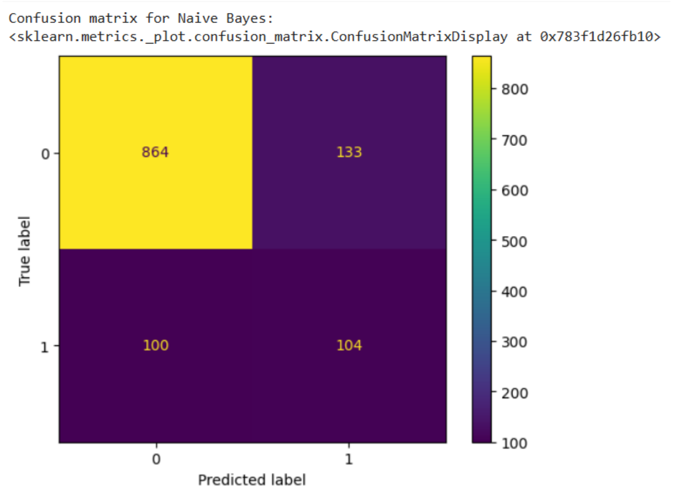
```
ensemble_learner.fit(X_train, y_train)
```

ii. In this analysis report,
1) paste the test confusion matrices, **2) AUC-ROC Curves** and **3) the classification reports** for each of the TWO base learners you chose to combine,
as well as **4) the test confusion matrix, 5) classification report for the voting Ensemble Learner** and **6) the AUC-ROC curve for the ensemble learner (optional).**
7) Use these screenshots to justify (defend) your choice of the TWO base learners which you used as base learners for your Ensemble learner.

1) Paste the test confusion matrices for each base learner (2 x matrices)

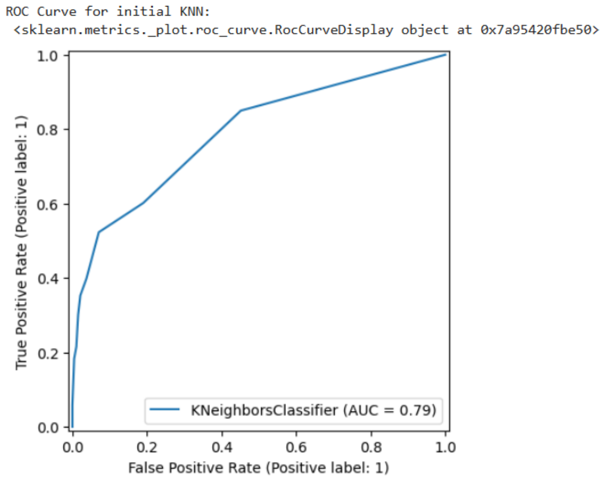


```
Confusion matrix for initial KNN:
```

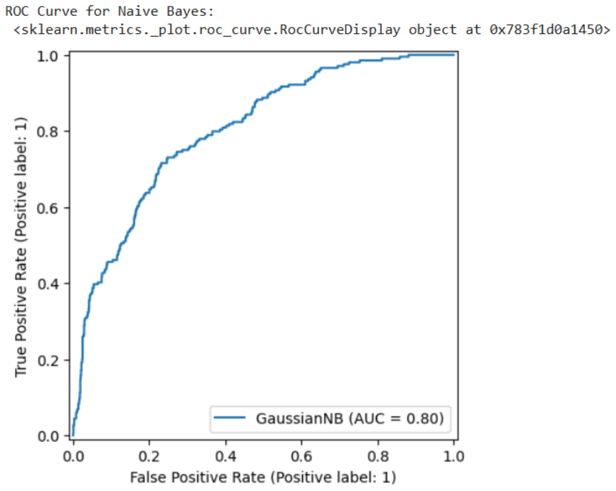


```
Confusion matrix for Naive Bayes:
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x783f1d26fb10>
```

2) Paste the AUC-ROC for each Base learner (2 x AUC-ROC graphs)

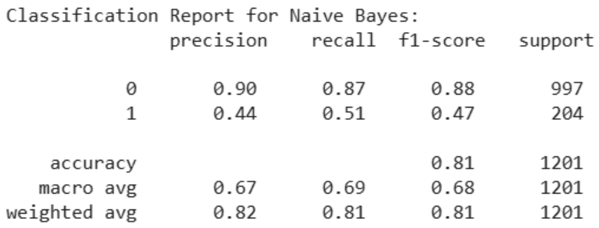


```
ROC Curve for initial KNN:
 <sklearn.metrics._plot.roc_curve.RocCurveDisplay object at 0x7a95420fbe50>
```

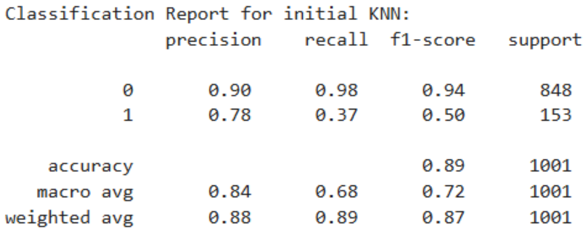


```
ROC Curve for Naive Bayes:
 <sklearn.metrics._plot.roc_curve.RocCurveDisplay object at 0x783f1d0a1450>
```

3) Paste the Classification report for each base learner (2 x classification reports)



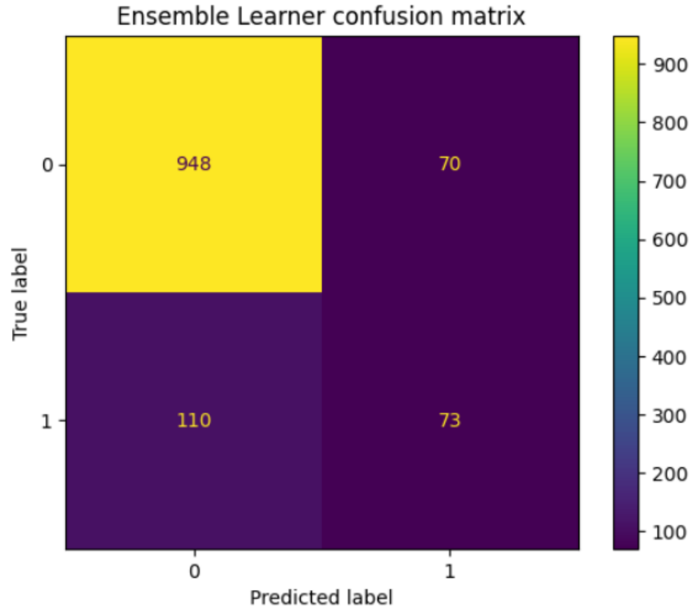
```
Classification Report for Naive Bayes:
              precision    recall  f1-score   support

           0       0.90      0.87      0.88       997
           1       0.44      0.51      0.47       204

    accuracy                           0.81      1201
   macro avg       0.67      0.69      0.68      1201
weighted avg       0.82      0.81      0.81      1201
```



```
Classification Report for initial KNN:
              precision    recall  f1-score   support

           0       0.90      0.98      0.94       848
           1       0.78      0.37      0.50       153

    accuracy                           0.89      1001
   macro avg       0.84      0.68      0.72      1001
weighted avg       0.88      0.89      0.87      1001
```

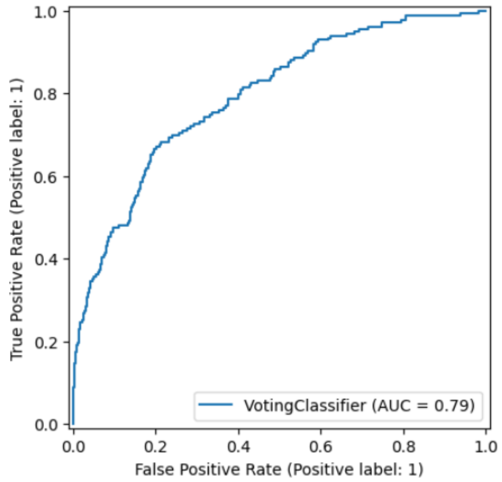
4) Paste the test confusion matrix for the ensemble learner (1 x confusion matrix)

```
Text(0.5, 1.0, 'Ensemble Learner confusion matrix')
```

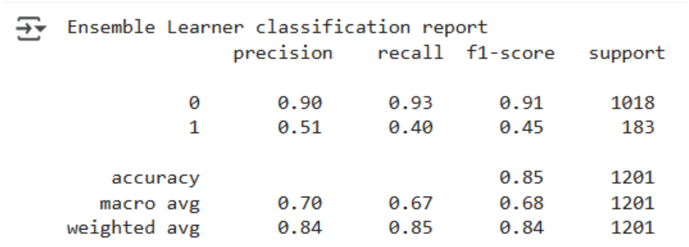


5) Paste the AUC-ROC for the ensemble learner (optional) (1 x AUC-ROC graph)

```
<sklearn.metrics._plot.roc_curve.RocCurveDisplay at 0x7a9541bc4650>
```



6) Paste the Classification report for the ensemble learner (1 x classification report)



```
Ensemble Learner classification report
              precision    recall  f1-score   support

           0       0.90      0.93      0.91      1018
           1       0.51      0.40      0.45       183

    accuracy                           0.85      1201
   macro avg       0.70      0.67      0.68      1201
weighted avg       0.84      0.85      0.84      1201
```

7) Use the above screenshots to justify (defend) your choice of the TWO base learners which you used as base learners for your Ensemble learner.
Using Naive Bayes and KNN, I am combining the weakest of the 3 models. The KNN model is strong on precision (0.85) and F1 score (0.73), and Naive Bayes has a good overall performance, especially in terms of AUC Roc (0.80). These two models complementary strengths could improve the ensemble's ability to handle the data and improve accuracy and precision.

ii. Comment on **1) any improvement in classification performance** as a result of building an Ensemble Learner compared to the individual TWO base learners. **2) Decide whether to recommend your ensemble learner for mortality prediction or one of the TWO base learners; 3) justify your recommendation.**

1) Accuracy: The ensemble learners accuracy is slightly lower than the KNNs accuracy but it is still higher than NB
KNN:0.88 NB:0.81 Ensemble learner: 0.86

The macro average and weighted average for the ensemble learner are better than NB, but the ensemble learner doesn't outperform KNN on these averages suggesting KNN is the top performer
Macro Average:
KNN:0.82 NB:0.64 Ensemble learner: 0.68
Weighted Average:
KNN:0.87 NB:0.82 Ensemble learner: 0.84

2) I would recommend KNN for mortality prediction as the models higher performance in precision and accuracy makes it a stronger choice overall

3) KNN offers the highest accuracy and precision for predicting breast cancer mortality, making it the best choice for ensuring high-quality predictions and minimising false positives (incorrectly predicting a patient as dead).
While the ensemble learner improves the Naïve Bayes model, the KNN's performance is stronger.
Naïve Bayes underperforms compared to both models, especially in terms of precision.

Case Study Title

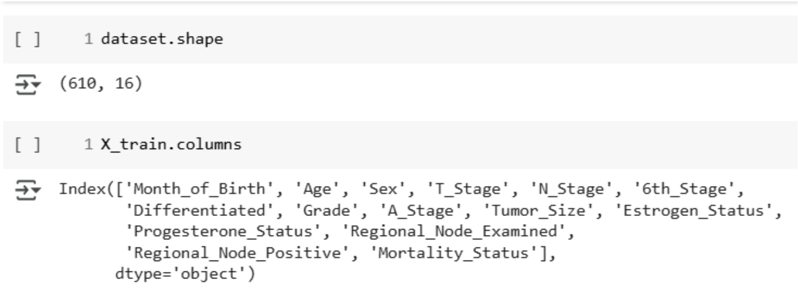
Case Study (B): Predicting Cancer Patients Survival Months.
Research Question: Does machine learning have the potential to assist doctors in predicting survival months for patients who are not going to survive breast cancer?

[Total 36 Marks]

Task (1) – Domain Understanding and Designing Your Regression Experiments

The healthcare professionals decided that regression modelling is required to predict survival months for those who would not survive breast cancer. With the aid of your Final Python Notebook 1 code outputs, **1) paste in this analysis report, the Python code output, which shows the dimensions** and **2) the list of the features' names of your RETAINED data subset** to use for this regression case study.

[2 Marks]



```
[ ]   dataset.shape
(610, 16)

[ ]   X_train.columns
Index(['Month_of_Birth', 'Age', 'Sex', 'T_Stage', 'N_Stage', '6th_Stage',
       'Differentiated', 'Grade', 'A_Stage', 'Tumor_Size', 'Estrogen_Status',
       'Progesterone_Status', 'Regional_Node_Examined',
       'Regional_Node_Positive', 'Mortality_Status'],
      dtype='object')
```

Task (2) – Modelling: Build Predictive Regression Models

a) Your healthcare team decided to use a decision tree regression (DT) algorithm to model the survival months. In less than 150 words, explain some added benefits of using a DT regressor to this healthcare prediction problem.

[2 Marks]

A decision tree regressor is useful for predicting survival months in breast cancer patients as it is highly interpretable and allows healthcare professionals to understand factors influencing predicted survival months easily. It captures complex relationships between patient variables and survival time and is easy to interpret and understand. This makes it ideal for situations such as in healthcare, where you need to explain how the model arrived at a prediction. It also is a clear visualisation, which makes it easy to understand how decisions are made, especially for stakeholders without a data background. This transparency is crucial in healthcare, where explaining model decisions is essential.
However, decision trees can be prone to overfitting, so a technique like pruning may be needed.

References with in-text citation

upGrad, no date. 'Pros and cons of decision tree regression in machine learning.' Available at: https://www.upgrad.com/blog/pros-and-cons-of-decision-tree-regression-in-machine-learning/ [Accessed 28/03/2025].

Task (2) – Modelling: Build Predictive Regression Models

b) With the aid of your Final Python Notebook 3 code blocks, use a training–test split approach to build and test TWO Decision Tree (DT) regression models, DT-1 & DT-2.

[6 Marks]

i. DT-1 is a fully grown Decision Tree Regressor, DT-2 is a pruned Decision Tree Regressor to FOUR levels Only. **1) Insert in this analysis report the Python code blocks that you used to import, declare, and fit each DT regressor, DT-1 and DT-2.**

```
from sklearn.tree import DecisionTreeRegressor
from sklearn import tree
X = dataset.drop('Survival_Months', axis=1)
y = dataset['Survival_Months']

X_train,X_test,y_train,y_test = train_test_split(X, y, test_size=0.3, random_state=42)
dt1_regressor = DecisionTreeRegressor()
dt1_regressor.fit(X_train, y_train)

dt1_figure = plt.figure(figsize=(200,200))
dt1_Graph = tree.plot_tree(dt1_regressor, feature_names=list(X_train.columns), filled=True)
dt1_figure.savefig("decision_tree.svg")

dt2 = DecisionTreeRegressor(max_depth=4, random_state=42)
dt2= dt2.fit(X_train, y_train)
y_pred_dt2 = dt2.predict(X_test)

dt2_figure = plt.figure(figsize=(200,200))
dt2_graph = tree.plot_tree(dt2, feature_names=list(X_train.columns), filled=True)
dt2_figure.savefig("pruned_decistion_tree.svg")
```

ii. Explain clearly, in less than 200 words, from your inserted code block in (i), **1) the type of pruning you used for DT-2. 2) Explain some of the benefits and disadvantages of the pruning method you used** in the context of (relation to) your cancer patients' regression modelling.

1) The pruning method is called pre-pruning; the tree has a maximum of four levels.

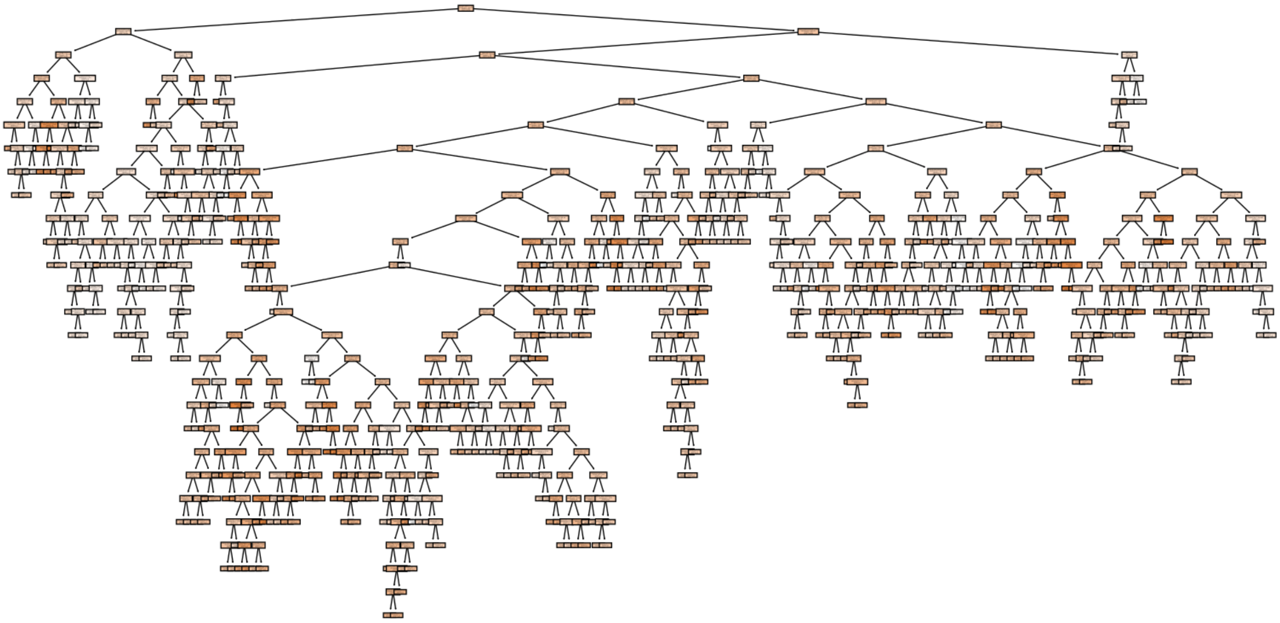
2) A benefit of pruning is that it reduces overfitting, which makes it more reliable in healthcare prediction.
It also improves interpretability, as a more straightforward tree will allow healthcare professionals to easily understand how predictions are made, helping transparent decision-making and passing this information onto patients.
A disadvantage to pruning is that there could be potential underfitting as some important survival patterns could be lost, and the model might not capture all complex relationships in the data. This could lead to the model missing some patterns, leading to less accurate predictions
Restricting the branches to the same depth might prevent the model from making detailed predictions.
There are also ethical issues as there is a potential for bias, where if the data that the models trains with has existing societal biases, the models can amplify those biases, leading to unfair outcomes. There is also a lack of transparency and accountability.

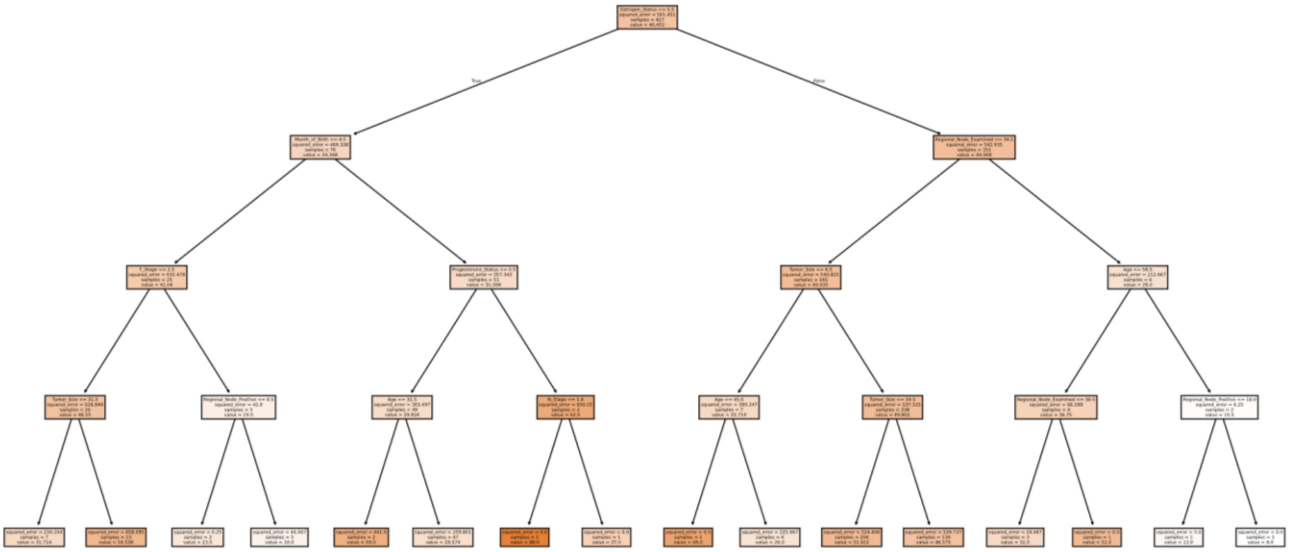
Task (2) – Modelling: Build Predictive Regression Models

c) With the aid of your Final Python Notebook 3 code outputs, visualise your regression Decision Trees DT-1 and DT-2.
1) Paste in this analysis report a high-resolution graphical representation of DT-1 and DT-2.

[4 Marks]

DT 1





Task (3) – Evaluating your Cancer Survival Months DT Regression Models

Your healthcare professionals provided the following success criteria to guide you when evaluating your DT-1 and DT-2 models. *"When evaluating both models' performances, which addresses your research question (b), the model is expected to make some errors in estimating the survival months. However, since the survival months calculations are made to try to save the lives of those who may not survive cancer by prioritising their treatment plans, it is important that the selected model signifies even small errors in survival months predictions."*

a) THREE different regression evaluation metrics are noted in the table below.
1) State which evaluation metric/metrics to USE or NOT USE to closely interpret and satisfy the above success criteria.
2) Justify (Defend) your choice of USE or DO NOT USE for each metric.
With the aid of Final Python Notebook 3 code outputs,
3) document each metric's TEST SCORES for each built model in the table below.

[8 Marks]

Metric	**1) USE or DO NOT USE**	**2) Justification for choosing "USING" or "NOT USING" this metric in relation to the success criteria**	**Model**	**3) Metric Test Score**
MSE	USE	MSE measures the average squared difference between the predicted and actual values (error). Since healthcare professionals need to detect errors, MSE is a metric to consider as it encourages the model to minimise errors as much as possible, and it is important to prioritise accurate predictions to inform appropriate treatment plans	DT-1 (Fully Grown)	979.9398907103825
			DT-2 (Pruned)	632.4444110089756
MAE	USE	MAE is the average of the absolute differences between predicted and actual values, which measures prediction which is essential when using a model to determine cancel patients survival months	DT-1 (Fully Grown)	23.87431693989071
			DT-2 (Pruned)	20.408353074167298
R-Square	DO NOT USE	R square shows the proportion of the variance of survival months (target variable). A negative R square values suggest that the model doesn't capture the variance well, hence it may not be the best metric to use when looking at accurately predicting survival months. This negative value could be due to a poor model or overfitting. However, in healthcare accuracy in models is more important than variance	DT-1 (Fully Grown)	-0.6131097011885136
			DT-2 (Pruned)	-0.04108652431881765

Task (3) – Evaluating your Cancer Survival Months DT Regression Models

b) **1) Suggest a single best regression model (DT-1 or DT-2)** based on your 'USED' performance metric/s scores, which you defended in Task (3a). **2) Explain how your suggested model fulfils the success criteria.**

[4 Marks]

1) DT2 pruned model is the best regression model. DT2 performs better than DT2, the fully grown decision tree, in both MSE and MAE, which shows that the pruned model has smaller squared errors and, on average, the pruned model predictions are closer to the actual values of survival months. These metrics align with the healthcare goal of reducing errors in predicting survival months and the prioritisation of treatment plans.

2) DT2 meets the success criteria as it prioritises accurate predictions, which is critical for ensuring that cancer patient treatment plans are prioritised correctly. Additionally, DT2 avoids complex decisions that could to larger errors when predicting survival months which ensures stability. The pruned tree is also more interpretable and allows healthcare professionals to easily understand how decisions are made and why certain survival months are being predicted.

Task (3) – Evaluating your Cancer Survival Months DT Regression Models

c) Describe to your healthcare team **any concerns you have about your selected performance metric/s** that you used to select your best decision tree model, which satisfies the success criteria. [200 words maximum] with in-text citation.

[4 Marks]

There are some concerns regarding the interpretability of these metrics, especially in a healthcare context where even small errors can be critical for patient care. With the mean absolute error (MAE), it is 20.41. This means that the model's predictions are off by approximately 20 months. This could be substantial when prioritising treatment as patients may not receive timely treatment, which could be detrimental.

The MSE for DT2 is 632.44, which shows the average square difference between actual and predicted values. This number suggests some predictions are significantly wrong. This significant deviation could lead to misclassifying patients who need immediate care, which could have serious consequences. As noted by Analytics Vidhya (2024), MSE is sensitive to large errors, and as you square the error, this metric may exaggerate the impact of its outliers.

While DT2 is the best model out of the two as it improves the accuracy of predictions, it still presents concerns regarding reliability, which can negatively impact patient treatment and care if not managed carefully.

References with in-text citation

Analytics Vidhya, 2024. Mean Squared Error: What is MSE and How Does it Work?
Available at: https://www.analyticsvidhya.com/blog/2024/07/mean-squared-error/ [Accessed 28 March 2025].

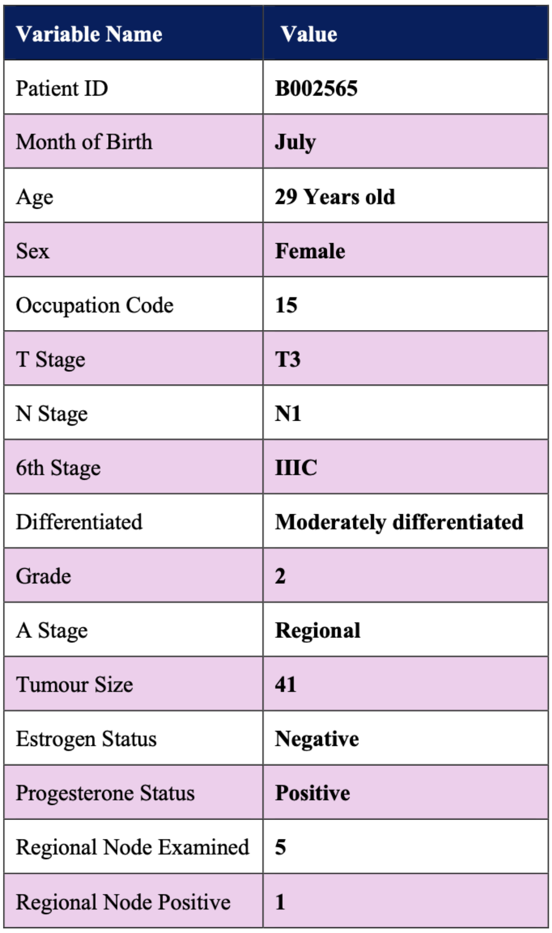
Task (4) – Interpreting Cancer Survival Months Decision Tree Outcomes

a) Patient B002565 breast cancer was deemed terminal. With the aid of your Final Python Notebook 3 outputs, **1) use your high-resolution graphical representation of your selected best DT regression model from Task (3.b) to predict the survival months for breast cancer patient B002565**;

2) you must write down which regression Decision Tree you used (DT-1 or DT-2) to estimate the survival months,

3) you must write down the path of rules (decision steps/tests) you used from your selected best DT to explain to patient B002565 how you estimated their predicted survival months. Patient B002565 attributes' values are in the following table:

[6 Marks]



Variable Name	Value
Patient ID	**B002565**
Month of Birth	**July**
Age	**29 Years old**
Sex	**Female**
Occupation Code	**15**
T Stage	**T3**
N Stage	**N1**
6th Stage	**IIIC**
Differentiated	**Moderately differentiated**
Grade	**2**
A Stage	**Regional**
Tumour Size	**41**
Estrogen Status	**Negative**
Progesterone Status	**Positive**
Regional Node Examined	**5**
Regional Node Positive	**1**

1) Predicted survival months: 37

2) I used DT2 to estimate the survival months for patient B002565

3) In decision trees, rules are the decision points the tree uses to make a prediction. We interpret the prediction for this patient by following the path of rules down the tree. Each decision is a split based on the variables and the patients values. At each step we check the patients values for that variable and follow that branch until we come to the decision outcome called a leaf node, which has the predicted survival month value.

The first decision node is Estrogen_Status <= 0.5 and the patients value is 0 therefore we go left (true). The next decision node is Month_of_Birth <= 4.5. The patients value is 7 therefore we go right (false). We land at the leaf node which gives us the value of 37.0 as the survival month decision.