UNIVERSITY OF WESTMINSTER

INFORMATICS INSTITUTE OF TECHNOLOGY

INFORMATICS INSTITUTE OF TECHNOLOGY

In Collaboration with

UNIVERSITY OF WESTMINSTER, UK

Machine Learning & Data Mining

5DATA001C.2

Analyses Report of Two Case Studies, Predicting Cancer Patients Mortality Status and Survival Months

Module Leader: Mr. Nipuna Senanayake

Name: C.M.M.S. Silva

UOW Id: W2053190

IIT Id: 20230183

Tutorial Group: CS 24

Case Study (A): Analyses Report for Predicting Cancer Patients Mortality Status Tasks.

Task (1) – Domain Understanding: Classification

Variable Name	RETAIN or DROP	Brief justification for retention or dropping
Patient ID	Drop	Patient ID is an identifier, not a predictive feature. It holds no clinical meaning. (PubMed Central, 2020)
Month of Birth	Drop	Not clinically relevant for breast cancer mortality. Seasonal birth has no known correlation.
Age	Retain	Age is a well-established risk factor for breast cancer progression and outcomes. (American Cancer Society, 2020)
Sex	Retain	Almost all breast cancer patients are female, but it's still important if there's any male presence.
Occupation	Drop	Occupation is often missing or too broad, and it's not directly related to mortality.
T Stage	Retain	Describes tumor size/spread in breast — critical for staging and outcome prediction.
N Stage	Retain	Indicates lymph node involvement, which is a strong predictor of mortality.
6th Stage	Retain	This is the overall TNM staging from American Joint Committee on Cancer 6th edition and it is highly predictive of outcomes. (AJCC, 2002)
Differentiated	Retain	Describes how much tumor cells differ from normal cells in grading. This affects prognosis.
Grade	Retain	Tumor grade reflects aggressiveness; higher grades often mean poorer prognosis.
A Stage	Retain	This is kind of another staging variable, potentially duplicative, but can be encoded carefully to test predictive value.
Tumor Size	Retain	Larger tumors generally have worse prognosis. And it is well supported in oncology literature. (CCS, 2023)
Estrogen Status	Retain	Hormone receptor status (ER) affects survival and treatment. Which is a crucial clinical marker. (NIH, 2021)
Progesterone Status	Retain	Like ER, PR status is highly predictive of survival and response to therapy.
Regional Node Examined	Retain	This indicates how thoroughly lymph nodes were assessed; useful for staging completeness.
Regional Node Positive	Retain	Indicates metastasis to lymph nodes. This is highly predictive of recurrence and mortality.
Survival Months	Retain	No need to drop it in the cleaning and preparing stages. May need to drop it in the modelling phases.
Mortality Status	Retain	This is the target variable for classification modelling of Breast Cancer Mortality.

Task (2) – Exploring and Understanding Your Dataset

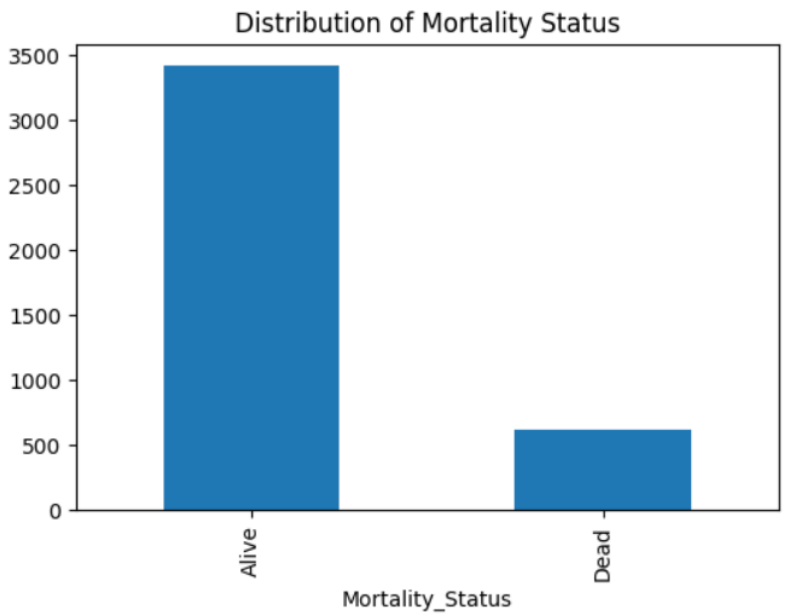
Descriptive Stats

	Age	Sex	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
count	4015.000000	4020	4024	4024	4024	4024	4024.000000	4024	4021.000000	4024	4024	4023.000000	4024.000000	4024.000000	4024
unique	NaN	2	4	3	5	4	NaN	2	NaN	2	2	NaN	NaN	NaN	7
top	NaN	Female	T2	N1	IIA	Moderately differentiated	NaN	Regional	NaN	Positive	Positive	NaN	NaN	NaN	Alive
freq	NaN	4001	1786	2732	1305	2351	NaN	3932	NaN	3755	3326	NaN	NaN	NaN	3399
mean	54.107098	NaN	NaN	NaN	NaN	NaN	2.150596	NaN	30.419299	NaN	NaN	14.373602	4.158052	71.472167	NaN
std	11.715528	NaN	NaN	NaN	NaN	NaN	0.638234	NaN	21.161080	NaN	NaN	8.129293	5.109331	25.361855	NaN
min	-50.000000	NaN	NaN	NaN	NaN	NaN	1.000000	NaN	-75.000000	NaN	NaN	1.000000	1.000000	1.000000	NaN
25%	47.000000	NaN	NaN	NaN	NaN	NaN	2.000000	NaN	16.000000	NaN	NaN	9.000000	1.000000	56.000000	NaN
50%	54.000000	NaN	NaN	NaN	NaN	NaN	2.000000	NaN	25.000000	NaN	NaN	14.000000	2.000000	73.000000	NaN
75%	61.000000	NaN	NaN	NaN	NaN	NaN	3.000000	NaN	38.000000	NaN	NaN	19.000000	5.000000	90.000000	NaN
max	502.000000	NaN	NaN	NaN	NaN	NaN	4.000000	NaN	140.000000	NaN	NaN	61.000000	46.000000	760.000000	NaN

Scale Types

```
Age                       float64
Sex                        object
T_Stage                    object
N_Stage                    object
6th_Stage                  object
Differentiated             object
Grade                       int64
A_Stage                    object
Tumor_Size                float64
Estrogen_Status            object
Progesterone_Status        object
Regional_Node_Examined    float64
Reginol_Node_Positive       int64
Survival_Months             int64
Mortality_Status           object
dtype: object
```

Target Distribution



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

Variable Name	Issue Found	Proposed Fix	Justification for used fix method
Age, Sex, Tumor_Size, Regional_Node_Examined, Occupation	Missing values	Use mean imputation to replace missing values for Age,Tumor_size and Regional_Node_Examined. Use mode imputation for Sex. Drop occupation column.	Using mean or median is common for numerical data. Mode is appropriate for categorical fields like Sex. Since Occupation's most values are missing and it's data is not useful for modeling, logical solution is to drop it.
Data type mismatch	No critical variables detected	No action required	All columns have appropriate data types for their values.
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	Duplicated Values	Drop the duplicated rows.	Duplicate rows represent redundant information and they do not contribute meaningfully to the analysis or model training.
Age, Tumor-Size	Negative Values	Replaced them with the mean of positive numerical values	Dropping them would result in loosing important data so replacing them with mean of median of positive value is the best option.

b)

Issue - Missing values

Before

```
Patient_ID                0
Month_of_Birth            0
Age                       9
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
Reginol_Node_Positive     0
Survival_Months           0
Mortality_Status          0
dtype: int64
```

After

```
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
Reginol_Node_Positive     0
Survival_Months           0
Mortality_Status          0
dtype: int64
```

Issue - Duplicated values

Before

```
1
```

After

```
0
```

Issue – Negative Numerical values

Before

```
Negative values found in column 'Age':
      Age
212 -50.0
--------------------
Negative values found in column 'Tumor_Size':
     Tumor_Size
210       -75.0
--------------------
```

After

```
No negative values found in column 'Age'
No negative values found in column 'Grade'
No negative values found in column 'Tumor_Size'
No negative values found in column 'Regional_Node_Examined'
No negative values found in column 'Reginol_Node_Positive'
No negative values found in column 'Survival_Months'
```

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

a)

Algorithm Name	Algorithm Type	Learnable Parameters	Some Strategic Hyperparameters
NB	Parametric	Prior probabilities and likelihoods for each feature and class	Regularization type Regularization strength
LR	Parametric	Weights and bias term	Variance smoothing (penalty, c, solver)
KNN (N=?)	Non- Parametric	Stores the training data	Number of neighbours Weight functions

b)

1) **Feature names**

```
Index(['Age', 'Sex', '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'],
      dtype='object')
```

Data shape

```
(4023, 15)
```

2)

A common choice for the training-test split ratio is 80:20, where 80% of the data is used for training and 20% for testing. This ratio offers a good balance between having enough data for the model to learn effectively and enough data to evaluate its performance on unseen data (Raschka, 2019). In many scenarios, this division offers a reasonable starting point, allowing for adequate model training while enabling a reliable assessment of its generalization capabilities.
Ratschka, S. (2019). Model evaluation, model selection, and algorithm selection in machine learning.

3)

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
```

In this case 20% is the specified proportion of the dataset to include in the test split. Also using a fixed value like 42 for random_state, ensures reproducibility and controls the shuffling applied to the data before applying the split.
Using the same test set for all models ensures a fair and consistent comparison of their performance (Géron, 2019).
Stratified sampling is crucial for ensuring the same label ratio in training and test sets.
It splits the data in a way that preserves the proportion of samples for each class in both subsets and ensures that the training and test sets have a similar distribution of classes (Kohavi, 1995). This helps prevent bias in model training and evaluation, especially when dealing with imbalanced datasets.

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

a)

Confusion Matrixes for Naïve Bayes, Logistic Regression and K-Nearest Neighbours models

```
Confusion Matrix for Naïve Bayes Model:
 [[586  96]
 [ 62  61]]
Confusion Matrix for Logistic Regression Model:
 [[667  15]
 [ 69  54]]
Confusion Matrix for K-Nearest Neighbours Model:
 [[660  22]
 [ 80  43]]
```

Classification Reports for Naïve Bayes, Logistic Regression and K-Nearest Neighbours models

Naïve Bayes Classification Report:

	precision	recall	f1-score	support
0	0.90	0.86	0.88	682
1	0.39	0.50	0.44	123
accuracy			0.80	805
macro avg	0.65	0.68	0.66	805
weighted avg	0.83	0.80	0.81	805

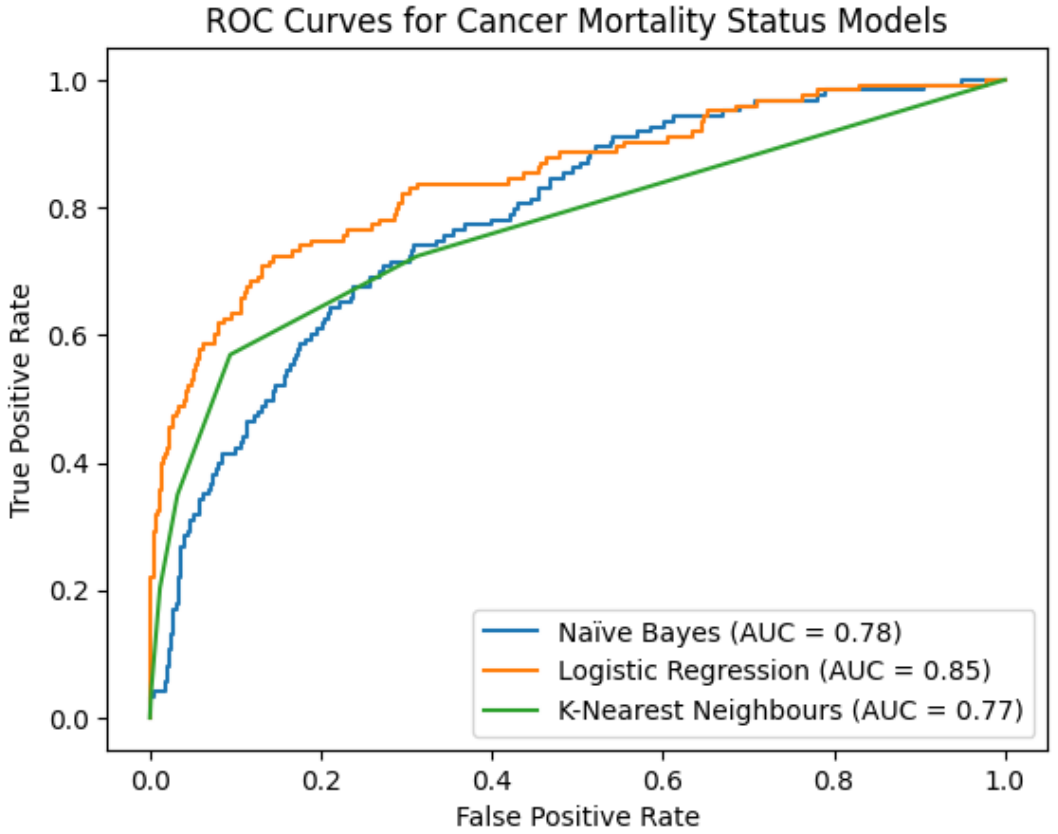
Logistic Regression Classification Report:

	precision	recall	f1-score	support
0	0.91	0.98	0.94	682
1	0.78	0.44	0.56	123
accuracy			0.90	805
macro avg	0.84	0.71	0.75	805
weighted avg	0.89	0.90	0.88	805

K-Nearest Neighbours Classification Report:

	precision	recall	f1-score	support
0	0.89	0.97	0.93	682
1	0.66	0.35	0.46	123
accuracy			0.87	805
macro avg	0.78	0.66	0.69	805
weighted avg	0.86	0.87	0.86	805

ROC curves for Naïve Bayes, Logistic Regression and K-Nearest Neighbours models



b)

Metrics	USE or DO NOT USE	Justification for choosing “USE” or “DO NOT USE” in relation to the success criteria	Model Name	Test Score
Accuracy	DO NOT USE	While accuracy gives an overall sense of model performance, it doesn't specifically address the need to discriminate between "Dead" and "Alive" patients, especially when class imbalance might exist. It might be misleading if one class is much larger than the other.	*NB*	0.803727
			LR	0.895652
			KNN (K=?)	0.873292
Recall	USE	Precision focuses on the accuracy of positive predictions. It's crucial for minimizing false positives, which aligns with the success criteria of better discrimination between classes. (e.g., predicting "Dead" when the patient is actually "Dead")	*NB*	0.803727
			LR	0.895652
			KNN (K=?)	0.873292
Precision	USE	Recall, also known as sensitivity, measures the ability to correctly identify all positive cases. It's important for minimizing false negatives, again aligning with the need for better discrimination. (e.g., identifying all "Dead" patients).	*NB*	0.825511
			LR	0.887358
			KNN (K=?)	0.856695
F-Score	USE	The F1-score is the harmonic mean of precision and recall, providing a balanced measure of both. It's useful when you want to consider both false positives and false negatives in your evaluation, contributing to better overall discrimination.	*NB*	0.813135
			LR	0.882965
			KNN (K=?)	0.856331
AUC-ROC	USE	The AUC-ROC represents the model's ability to distinguish between the two classes across different thresholds. A higher AUC indicates better discrimination capability, directly addressing the success criteria.	*NB*	0.782216
			LR	0.846244
			KNN (K=?)	0.769091

c) **Suggested best mortality status classification model – Logistic Regression Model**

Logistic Regression likely demonstrated a good balance between Precision and Recall, yielding a high F1-score indicating better overall performance in distinguishing between classes and a superior AUC-ROC curve consistently staying above those of Naïve Bayes and K-Nearest Neighbours (KNN), especially in the lower false positive rate region, which is critical for medical diagnoses. This indicates its effectiveness in discriminating between "Dead" and "Alive" cancer patients, satisfying the healthcare professionals' need for a model that minimizes both false positives and false negatives, leading to more reliable predictions for improved patient care.

d)

1) **Specifying a parameters grid and applying the GridSearchCV**

```
param_grid = {'penalty': ['l1', 'l2'],'C': [0.1, 1, 10], 'solver': ['liblinear', 'saga'] }
grid_search = GridSearchCV(LogisticRegression(max_iter=2000), param_grid, cv=5, scoring='accuracy')
grid_search.fit(X_train, y_train)
```

Best hyperparameters – c : 0.1, penalty : l2, solver : liblinear

```
Best Logistic Regression parameters: {'C': 0.1, 'penalty': 'l2', 'solver': 'liblinear'}
```

2) **Confusion Matrix**

Before Hyperparameter Tuning

```
Confusion Matrix for Logistic Regression Model:
 [[667  15]
 [ 69  54]]
```

After Hyperparameter Tuning

```
Confusion Matrix for Tuned Logistic Regression Model:
 [[667  15]
 [ 69  54]]
```

USED performance metrics scores

Performance Metrics	Scores Before Hyperparameter Tuning	Scores After Hyperparameter Tuning
Recall	0.895652	0.895652
Precision	0.887358	0.887358
F-Score	0.882965	0.882965
AUC-ROC	0.846244	0.846744

The confusion matrix values for the tuned Logistic Regression model are not significantly different from the original model's values, even after hyperparameter tuning. This may be due to issues like,

Limited Hyperparameter Search Space - The hyperparameter grid defined for GridSearchCV might not have included the optimal values for the dataset. Expanding the search space by adding more values or a wider range for the existing hyperparameters may work.

Insufficient Data - If the dataset is relatively small, the impact of hyperparameter tuning might be less noticeable. Acquiring more data if possible is a good solution, as this can often lead to more significant improvements in model performance.

The AUC-ROC score has slightly increased after hyperparameter tuning, from 0.846244 to 0.846744. Therefore, the tuned Logistic Regression model, with the slightly higher AUC-ROC, is likely to have a marginally better ability to discriminate between "Dead" and "Alive" cancer patients compared to the original model. This improvement, though small, could potentially translate to more accurate predictions and better patient care decisions.

e) When considering using the tuned Logistic Regression model for predicting breast cancer mortality status the following limitations and ethical issues should be taken into account and addressed.

Limitations

Data Dependence: The model's performance is heavily reliant on the quality and representativeness of the training data. If the data used to train the model is biased or incomplete, the model's predictions may be inaccurate or unreliable for certain patient populations.
Limited Generalizability: The model's performance may not generalize well to unseen data or different patient cohorts. Factors like changes in treatment protocols, advancements in medical technology, or variations in patient demographics can affect the model's accuracy over time and across different settings.
Lack of Interpretability: Although Logistic Regression provides some interpretability through its coefficients, understanding the complex interactions between features and their contribution to mortality risk can be challenging. This can limit the model's transparency and hinder clinicians' ability to fully trust and explain its predictions to patients.

Ethical Issues

Bias and Discrimination: If the training data reflects existing biases or disparities in healthcare access or treatment, the model might perpetuate these inequalities in its predictions, potentially leading to unfair or discriminatory outcomes for certain patient groups.

Overreliance and Deskilling: Overreliance on the model's predictions without adequate clinical judgment could potentially lead to deskilling of healthcare professionals and a decline in the quality of patient care. It's essential to view the model as a tool to support clinical decision-making, not as a replacement for human expertise.
Transparency and Explainability: The lack of complete transparency and interpretability of the model's predictions can create challenges in establishing trust and accountability. Ensuring that the model's logic and limitations are clearly communicated to both clinicians and patients is vital for ethical use.

f)

1)

```
#Importing necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import VotingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report, confusion_matrix, roc_curve, auc
```

```
# Based on Notebook 1 and Notebook 2 results, using Logistic Regression and Naive bayes for creat
lr = LogisticRegression(C=0.1, penalty='l2', solver='liblinear', max_iter=2000, random_state=42)
nb = GaussianNB()
```

```
# Create the ensemble voting classifier based on the probability
ensemble = VotingClassifier(estimators=[('lr', lr), ('nb', nb)], voting='soft')
```

```
# Train the ensemble
ensemble.fit(X_train, y_train)
```

2)

Naïve Bayes Model

Classification Report

```
Naïve Bayes Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.86      0.88       682
           1       0.39      0.50      0.44       123

    accuracy                           0.80       805
   macro avg       0.65      0.68      0.66       805
weighted avg       0.83      0.80      0.81       805
```

Confusion Matrix

```
Confusion Matrix for Naïve Bayes Model:
 [[586  96]
 [ 62  61]]
```

Logistic Regression Model

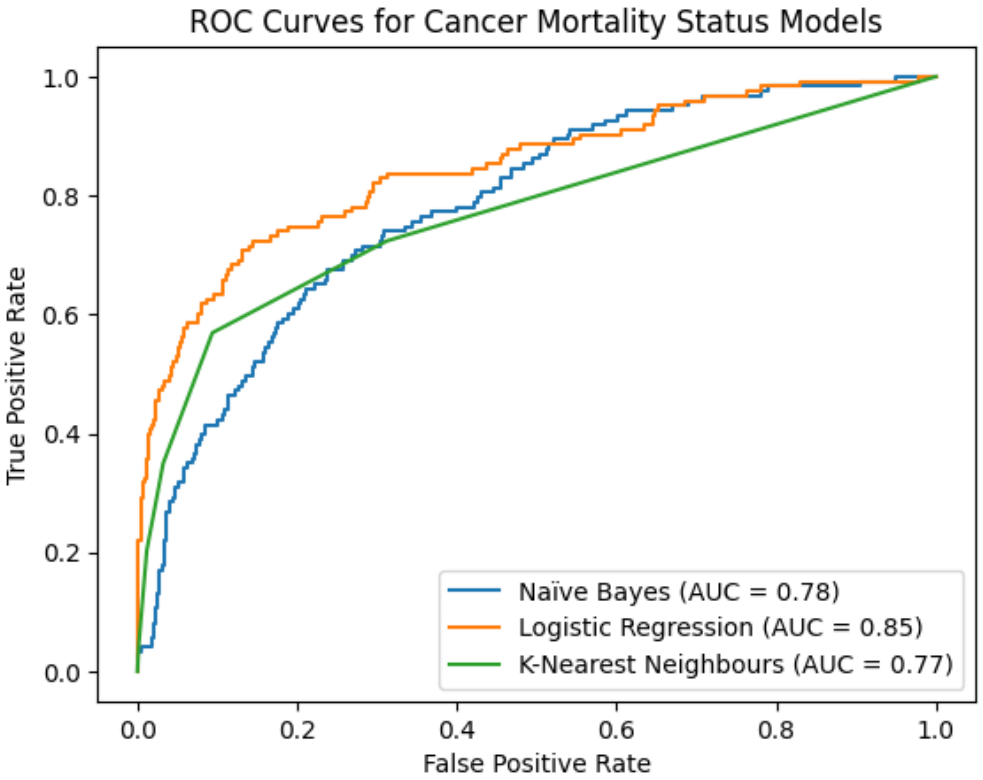
Classification Report

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

           0       0.91      0.98      0.94       682
           1       0.78      0.44      0.56       123

    accuracy                           0.90       805
   macro avg       0.84      0.71      0.75       805
weighted avg       0.89      0.90      0.88       805
```

Confusion Matrix

```
Confusion Matrix for Logistic Regression Model:
 [[667  15]
 [ 69  54]]
```

AUC-ROC Curves Two Base Learners



Confusion Matrix and Classification Report for the Voting Ensemble Learner

```
--- Ensemble Classifier Results ---

Confusion Matrix:
[[567 115]
 [ 71  52]]

Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.83      0.86       682
           1       0.31      0.42      0.36       123

    accuracy                           0.77       805
   macro avg       0.60      0.63      0.61       805
weighted avg       0.80      0.77      0.78       805
```

Justification for Choosing Logistic Regression and Naive Bayes:

Diverse Strengths: Logistic Regression and Naive Bayes have different strengths. Logistic Regression is a discriminative model that directly learns the decision boundary, while Naive Bayes is a generative model that learns the underlying data distribution. Combining these diverse models can help improve overall performance by capturing different aspects of the data.

Individual Performance: Both Logistic Regression and Naive Bayes showed reasonable performance on their own. They had decent accuracy scores, AUC-ROC values, and F1-scores. This suggests they have some predictive power for the target variable.

Ensemble Improvement: The ensemble learner, by combining Logistic Regression and Naive Bayes with soft voting, achieved a slight improvement in the weighted average F1-score compared to the individual base learners. This indicates that combining these models leads to better overall performance.

3)

Improvement in Classification Performance:

Ensemble Learner: By combining Logistic Regression and Naive Bayes using soft voting (probability-based), the ensemble aims to leverage the strengths of both models. This obviously leads to better generalization and potentially higher accuracy compared to individual models. In conclusion we can be satisfied that the ensemble truly performed better, considering the classification report or AUC value. improvement.

Base Learners (Logistic Regression and Naive Bayes): These individual learners might have limitations on their own. For example, Logistic Regression assumes linearity between the features and target, while Naive Bayes assumes feature independence. The ensemble method can help mitigate these weaknesses.

When considering evaluation metrics like AUC score, Precision, Recall and confusion matrix, especially true positive and true negative scores I recommend using the Logistic Regression Model.

Logistic Regression Model have already achieved high performance on above mentioned matrics and the ensemble doesn't give a considerable boost when comparing and it would be sufficient to go with the Logistic Regression Base model due to its simplicity and interpretability.

==========END OF CASE STUDY (A)==========

Case Study (B): Analyses Report for Predicting Cancer Patients Survival Months Tasks.

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

Shape of Regression dataset

```
(4023, 15)
```

Retained features for regression modeling

```
Index(['Age', 'Sex', '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'],
      dtype='object')
```

Task (2) – Modelling: Build Predictive Regression Models
a)

Decision Trees offer several advantages in predicting survival months:

Interpretability: Decision Trees are easy to understand and visualize, allowing healthcare professionals to grasp the factors influencing survival predictions.

Handling Non-linearity: They can capture complex, non-linear relationships between features and survival outcomes, unlike linear models.

Feature Importance: Decision Trees automatically rank features based on their importance in predicting survival, providing insights for medical research.

Robustness to Outliers: Decision Trees are relatively robust to outliers in the data, leading to more stable predictions.

b)

1)

```
# Create a fully grown Decision Tree (DT-1)
dt_full = DecisionTreeRegressor(random_state=42)
dt_full.fit(X_train, y_train)
```

```
# Create a pruned Decision Tree (DT-2) - limit to 4 levels
dt_pruned = DecisionTreeRegressor(max_depth=4, random_state=42)
dt_pruned.fit(X_train, y_train)
```

2)

Pruning Method:

For pruning I'm using pre-pruning by setting the **max_depth** parameter to 4. This limits the depth of the decision tree to four levels during the training process. It stops the tree from growing beyond this depth, effectively preventing it from learning very specific, potentially noisy patterns in the data.

Benefits in this Context:

Reduced Overfitting: Pre-pruning with **max_depth** prevents the model from becoming too complex and overfitting the training data, leading to better generalization to unseen cancer patient data.

Improved Interpretability: Limiting the tree's depth makes it smaller and easier for healthcare professionals to understand and interpret the factors influencing survival predictions.

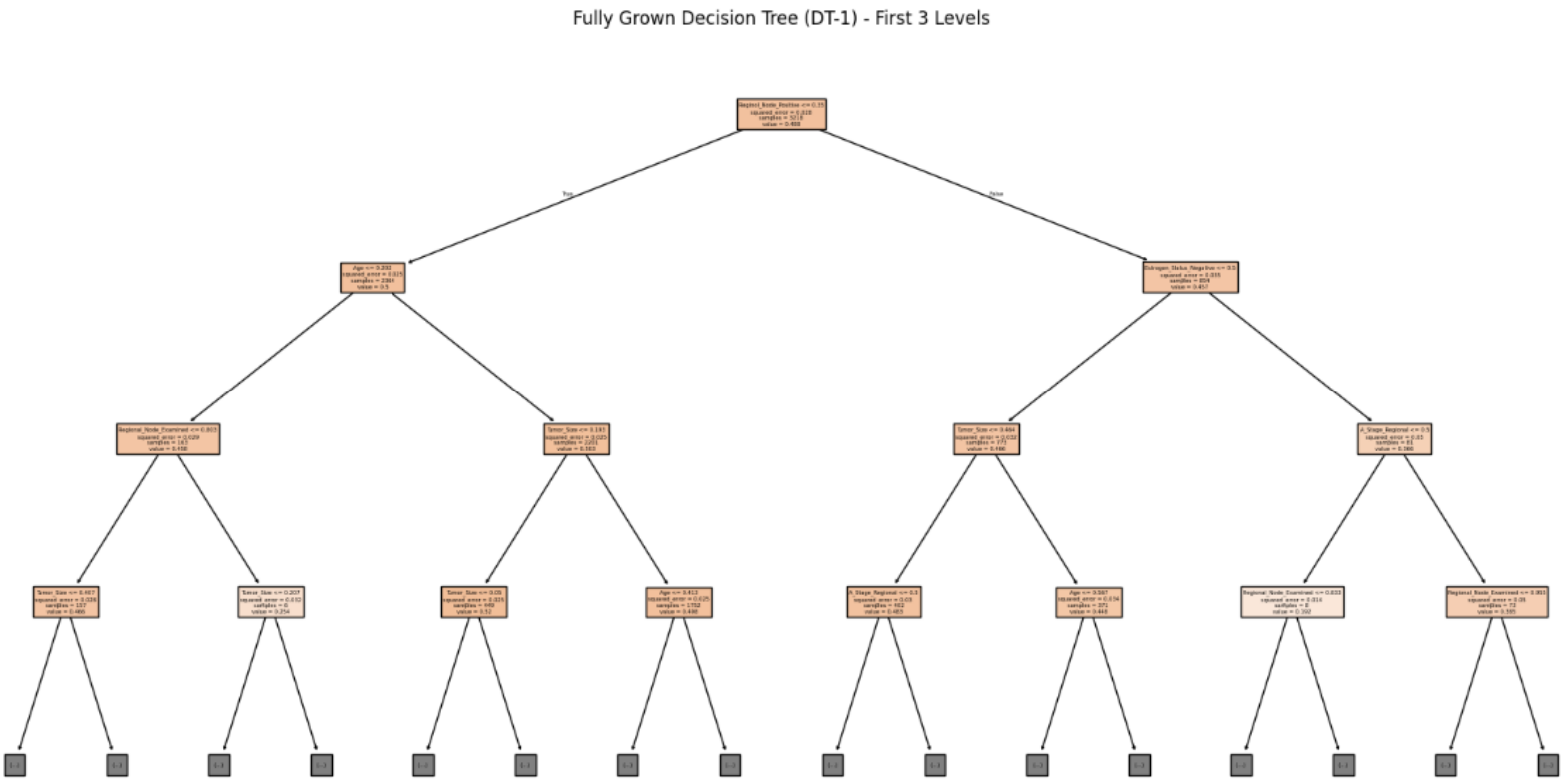
Faster Predictions: A smaller tree leads to faster predictions, which can be beneficial for time-sensitive medical decisions.

Disadvantages in this Context:

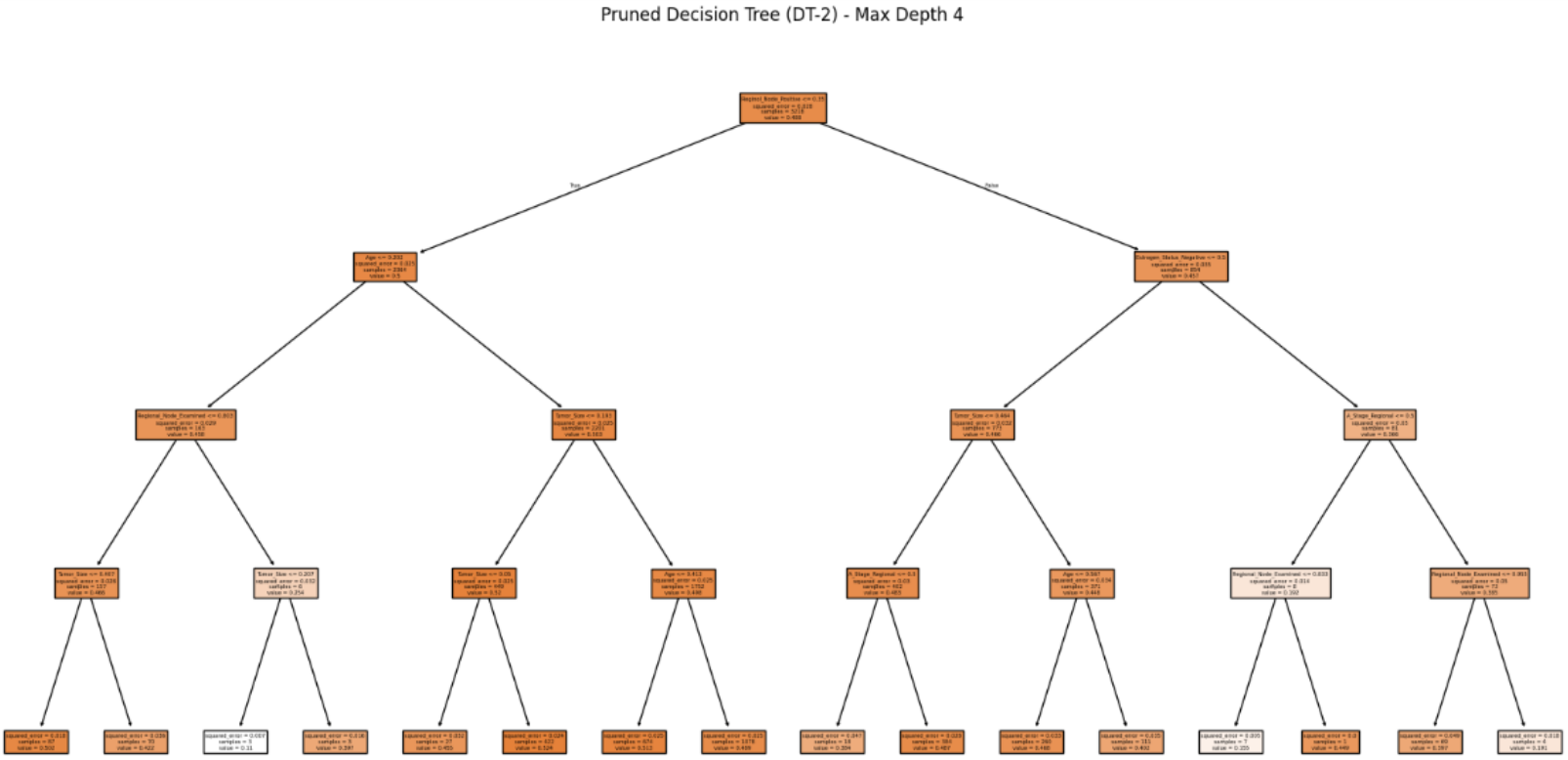
Potential Underfitting: If the optimal tree depth for accurate survival prediction is greater than 4, setting **max_depth** too low can lead to underfitting, where the model misses important patterns. This could result in less accurate survival estimates for some patients.

Limited Exploration: Pre-pruning restricts the tree's ability to explore deeper relationships in the data. It might miss subtle but potentially valuable interactions between features affecting cancer patient survival.

c) **Fully Grown Decision Tree (DT – 1)**



Pruned Decision Tree (DT – 2)



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

Metrics	USE or DO NOT USE	Justification in relation to the success criteria	Model Name	Test Score
MSE	USE	MSE penalizes larger errors more heavily due to the squaring. This aligns with the requirement that even small errors are significant, as larger errors would be amplified and highlighted by the MSE.	DT-1 (Fully Grown DT)	0.06
			DT-2 (Pruned DT)	0.03
MAE	USE	MAE provides a direct interpretation of the average error in survival months prediction. It's easy to understand and reflects the typical magnitude of errors made by the model. While it doesn't emphasize larger errors as much as MSE, it still gives valuable insight into the model's accuracy in predicting survival months.	DT-1 (Fully Grown DT)	0.19
			DT-2 (Pruned DT)	0.14
R-Square	DO NOT USE	R-squared primarily measures the overall goodness of fit of the model, indicating the proportion of variance in the dependent variable explained by the independent variables. While useful, it doesn't directly address the requirement of highlighting small errors in survival months predictions. A high R-squared doesn't necessarily mean the model is accurate for individual predictions, especially regarding smaller errors.	DT-1 (Fully Grown DT)	-0.90
			DT-2 (Pruned DT)	0.02

b)

Based on the 'USED' performance metrics (MSE and MAE), I recommend DT-2 (Pruned Decision Tree) as the single best regression model.

The healthcare professionals emphasized the importance of signifying even small errors in survival months predictions to prioritize treatment plans and potentially save lives. DT-2 fulfills this success criteria better than DT-1 due to the following reasons:

Sensitivity to Small Errors: By having lower MSE and MAE, DT-2 demonstrates better sensitivity to smaller errors in predictions. This is crucial for identifying patients who may require more immediate attention or altered treatment strategies based on potentially shorter survival times.

Reduced Risk of Larger Errors: The lower MSE of DT-2 specifically indicates a reduction in the occurrence of larger prediction errors. This is critical in a healthcare context where significant miscalculations can have serious consequences.

While both models likely have some degree of error in estimating survival months, DT-2's superior performance in terms of MSE and MAE suggests that it is better at capturing and highlighting smaller errors. This alignment with the healthcare professionals' success criteria makes DT-2 the more suitable choice for this specific task, where accuracy and sensitivity to smaller errors are paramount.

c)

While MSE and MAE are valuable metrics for evaluating regression models and were deemed suitable for this specific task, they do have some limitations and potential concerns that should be acknowledged.

Sensitivity to Outliers: Both MSE and MAE can be heavily influenced by outliers in the dataset. Extreme values in survival months could disproportionately affect these metrics, potentially leading to an overestimation of the model's error. This is especially relevant in healthcare data, where outliers might represent unique patient cases or data entry errors.

Equal Weighting of Errors: MAE treats all errors equally, regardless of their direction (overestimation or underestimation). In a healthcare context, underestimating survival months could have more severe implications than overestimation. The healthcare team should be aware that MAE might not fully reflect the potential consequences of different types of errors.

Limited Insight into Error Distribution: While these metrics provide an overall picture of the model's error, they don't offer detailed insights into the distribution of errors across different patient subgroups or survival time ranges. Further analysis might be needed to understand if the model performs differently for specific patient populations.

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

To estimate the predicted survival months for breast cancer patient **B002565**, I utilized the **pruned Decision Tree regression model DT-2**, which was selected as the best-performing model based on previous evaluations.
Using the patient's clinical attributes, traced the following decision path through the DT-2 model.
The patient has **1 positive regional lymph node**, which is greater than 0.5, so I proceed to the right child node. The **estrogen status is negative**, meeting the condition Estrogen Status_Negative <= 0.5, so I proceed to the left child node. The **tumor size is 41**, which satisfies the condition Tumor Size <= 64.0, leading us further left. The patient's **age is 29**, which is less than or equal to 44.5, so I move to the final left child node. At this terminal node, the model predicts a survival time of approximately **44.62 months**. This prediction is based on the path of rules matched specifically to patient B002565's profile and provides a data-driven estimate of their expected survival time.

==========END OF CASE STUDY (B)==========