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

Name: **K. A. D. Hasindu Amalka Pieris**

Table of Contents

Introduction

The purpose of this paper is to present an analysis report on two case studies which are predicting cancer patients' mortality status and survival months related to the three Python notebooks that were created for: data understanding, cleaning and preparation, mortality status classifiers, their performances and hyperparameters optimization, mortality status ensemble classifier with its base learners' performances and survival months regression decision trees, with their graphical representation and performances.

Case Study (A): Predicting Cancer Patients Mortality Status

Task (1) – Domain Understanding: Classification

Variable Name	RETAIN or DROP	Brief justification for retention or dropping
Patient ID	DROP	Identifier has no impact on the mortality status
Month of Birth	DROP	No impact on the mortality status
Age	RETAIN	Younger ages are associated with more subtypes of cancer. *(González et al., 2020)*
Sex	RETAIN	While breast cancer is more common in women, male breast cancer tends to be diagnosed at a later stage. *(Giordano et al., 2004)*
Occupation	DROP	Have no impact on breast cancer
T Stage	RETAIN	Higher T stage indicates larger tumor *(Gonzalez et al., 2007)*
N Stage	RETAIN	The presence of cancer in regional lymph nodes is a strong predictor of mortality *(Gonzalez et al., 2007)*
6^{th} Stage	RETAIN	AJCC 6^{th} staging is widely used for cancer staging*(AJCC, 2017)*
Differentiated	RETAIN	Higher grade tumors lead to higher mortality *(Schwartz et al., 2014)*
Grade	RETAIN	Rate of tumor growth
A Stage	RETAIN	Shows the distant metastasis *(AJCC, 2017)*
Tumor Size	RETAIN	Large tumors have higher risk *(Crowe et al., 1992)*
Estrogen Status	RETAIN	ER-positive tumors often have better outcomes *(Yu et al., 2021)*
Progesterone Status	RETAIN	PR positivity has favorable prognosis *(Diana et al., 2022)*

Regional Node Examined	RETAIN	Number of nodes impact on staging
Regional Node Positive	RETAIN	Number of positive nodes impact on survival *(Ohri et al., 2017)*
Survival Months	DROP	Continuous data and not necessary for binary outcome
Mortality Status	RETAIN	Target variable *(Alive vs Dead)*

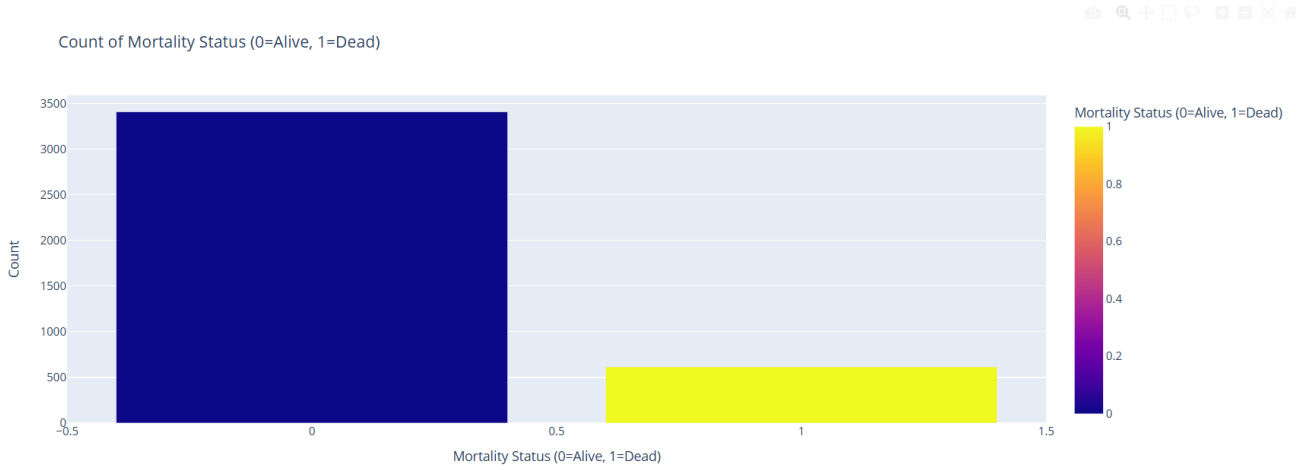
Task (2) – Exploring and Understanding Your Dataset

Descriptive Statistics for Numerical Variables:

	Age	Tumor_Size	Regional_Node_Examined	Regional_Node_Positive
count	4024.000000	4024.000000	4024.000000	4024.000000
mean	53.981362	26.694583	14.238072	4.151590
std	8.951489	14.659001	7.677341	5.066241
min	30.000000	1.000000	1.000000	1.000000
25%	47.000000	16.000000	9.000000	1.000000
50%	54.000000	23.000000	14.000000	2.000000
75%	61.000000	34.250000	19.000000	5.000000
max	69.000000	70.000000	34.000000	34.000000

```
Value Counts for Categorical Variables:

Sex:
Sex
0    4024
Name: count, dtype: int64

T_Stage:
T_Stage
1    1786
0    1603
2     533
3     102
Name: count, dtype: int64

N_Stage:
N_Stage
0    2732
1     820
2     472
Name: count, dtype: int64
```

```
Differentiated:
Differentiated
0    2351
1    1111
3     543
2      19
Name: count, dtype: int64

Estrogen_Status:
Estrogen_Status
1    3755
0     269
Name: count, dtype: int64

Progesterone_Status:
Progesterone_Status
1    3326
0     698
Name: count, dtype: int64

Mortality_Status:
Mortality_Status
0    3408
1     616
Name: count, dtype: int64
```



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

a)

Variable Name	Issue found	Proposed fix	Justification for used fix method
Age	9 NaN data and 4 outliers	Calculating the mean age and replacing the NaN data and outliers with it	This method was used since age cannot have extreme values (more than 100 or negative) and since only a few datasets missed age, to save them, mean was used
Sex	4 NaN data and few binary data	Found the mode and replaced NaN data and binaries with it	Since this is about breast cancer and obviously most of the time it affects to women, therefor 'Female' was used

Tumor Size	3 NaN data and 221 outliers	Found the mean tumor size and replaced NaN data and outliers with it	Tumor size cannot be extremely larger or negative or non. Therefor this method was used to conserve the data and prepare the datasets
Regional Node Examined	1 NaN data and 73 outliers	Found mean value and replaced NaN data and outliers with it	Having extreme number of nodes is highly unlikely and therefor this method was used to conserve the data and prepare the datasets
Survival Months	19 outliers	Found mean value and replaced outliers with it	Number of survival months from a breast cancer cannot be a very large value or a very small value and therefor this method was used to prepare the datasets
Regional Node Positive	Change name from “Reginol” to “Regional”	Rename the column	Renaming the column to make more sense about the data
Mortality Status	Having different ways of capitalization formats	Recapitalize the data	Recapitalizing whole dataset so it can be easily encoded

b)

- Renaming “Reginol Node Positive” into “Regional Node Positive”

Reginol_Node_Positive
1
5
7
1
1

Regional_Node_Positive
1
5
7
1
1

- Using mean for numerical columns and mode categorical columns to fill the NaN values

	0
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

```
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
```

- Replacing binary data in categorical column “Sex” with its mode

```
Unique values in 'Sex':
['Female' '1']
```

```
['Female']
```

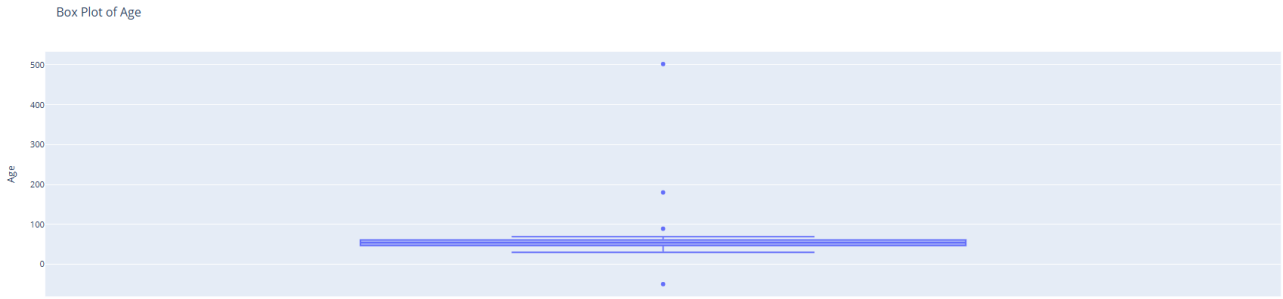
- Capitalizing mortality status dataset

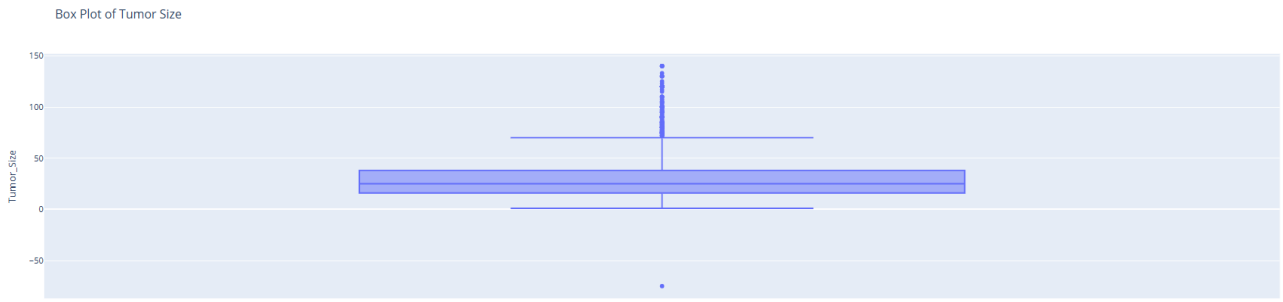
```
Unique values in 'Mortality_Status':
['Alive' 'Dead' 'ALIVE' 'DEAD' 'ALive' 'alive' 'dead']
```

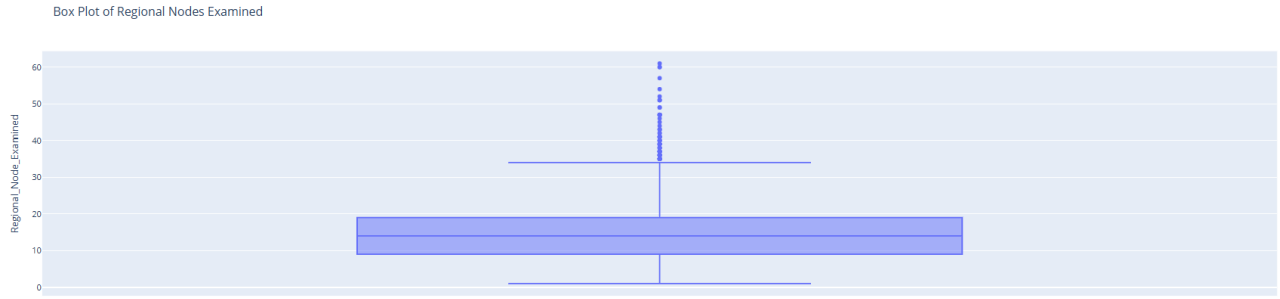
```
['Alive' 'Dead']
```

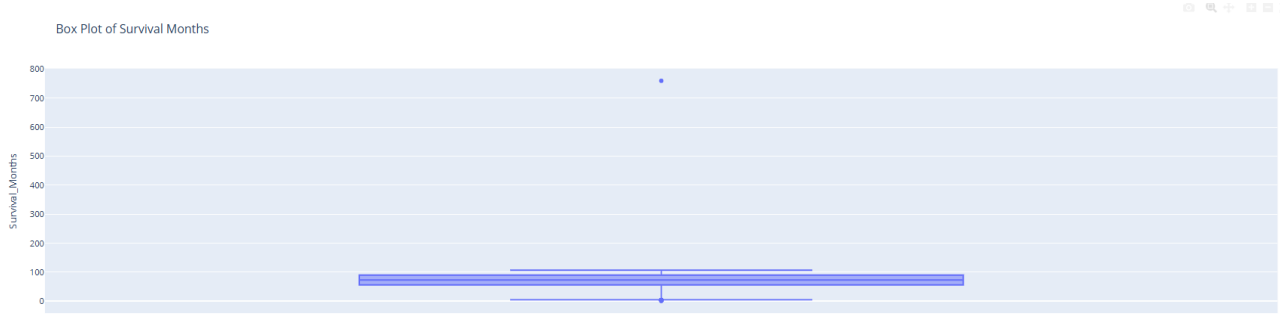
- Handling outliers

(Before Handling outliers)

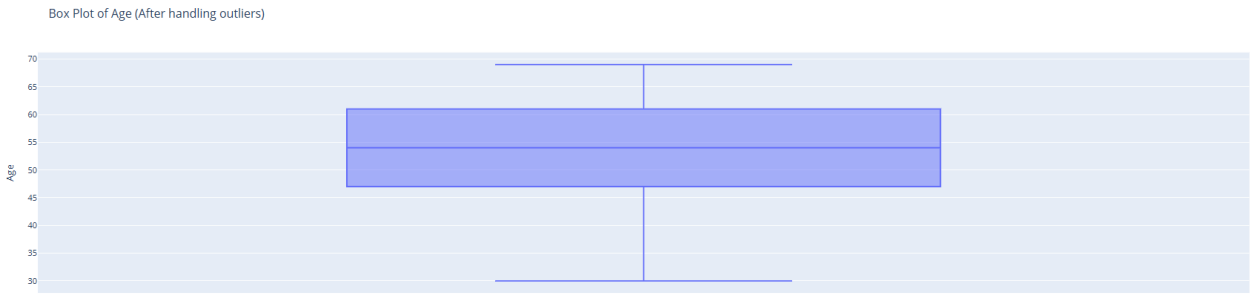




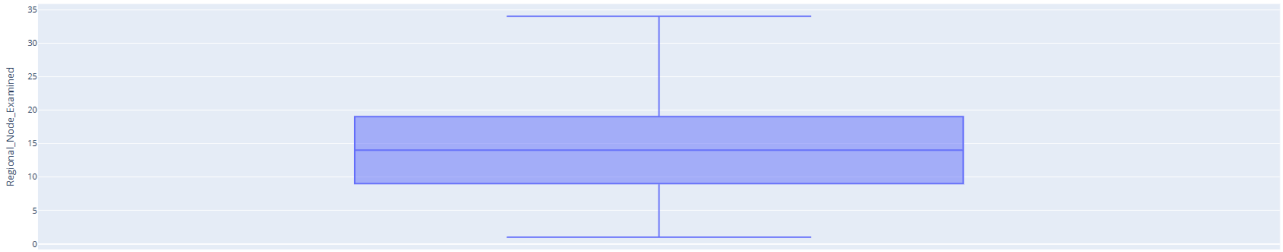




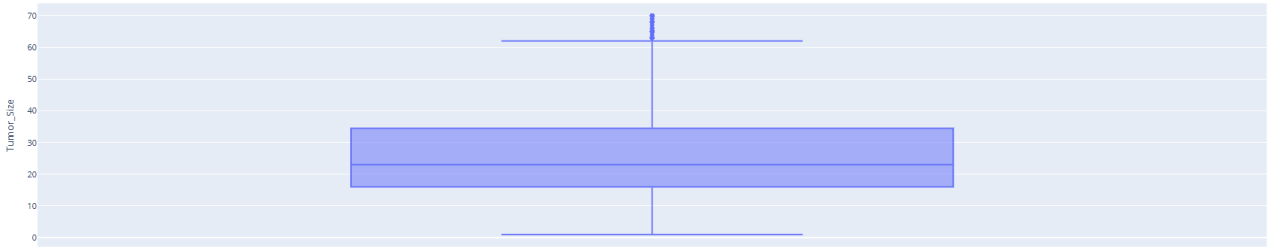
(After handling outliers)



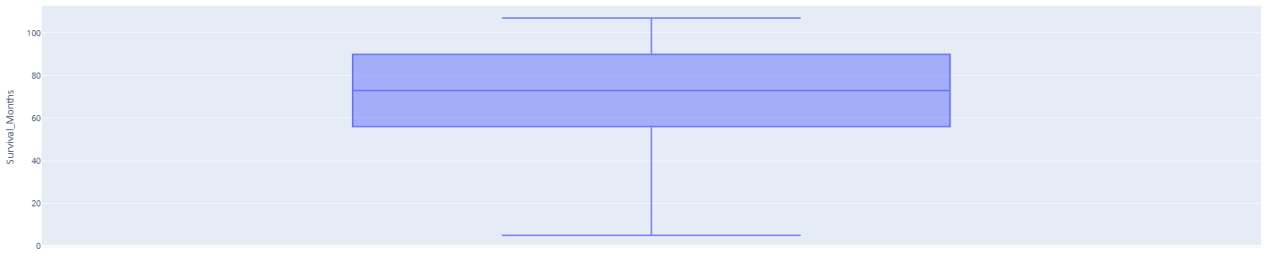
Box Plot of Regional Nodes Examined (After handling outliers)



Box Plot of Tumor Size (After handling outliers)



Box Plot of Survival Months



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

a)

Algorithm Name	Algorithm Type	Learnable Parameters	Some Strategic Hyperparameters
NB	Parametric	Class prior probabilities	Smoothing parameter
LR	Parametric	Coefficients and intercepts	Regularization type and regularization strength
KNN (N = 5)	Non-Parametric	None (Lazy learner)	Number of neighbors, distance metric

b)

i)

```
Index(['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')
```

```
(4024, 14)
```

ii)

For the **Final Python Notebook 2,** out of all the classification data, 80% was used for the training purpose and the remaining 20% was used for testing. An 80/20 training-test split is a widely used data split standard in machine learning since it has a reasonable balance between model training and evaluation. 80% of data used for the testing ensures that the models have enough information to learn and other 20% gives an unbiased data sample to test the trained models. Most of the time, this ratio of train and test prevents the models from overfitting and underfitting. Moreover, since mortality status is an imbalanced dataset, applying stratification ensures class distribution remains consistent across splits. 80/20 split often offers a practical trade-off between bias and variance in performance estimation. *(Raschka, 2018)*

iii)

```
#Spliting data for training(80%) and testing(20%)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

print(y_train.value_counts(normalize=True))

Mortality_Status
0    0.846847
1    0.153153
Name: proportion, dtype: float64

print(y_test.value_counts(normalize=True))

Mortality_Status
0    0.847205
1    0.152795
Name: proportion, dtype: float64
```

(Alive = 0 , Dead = 1)

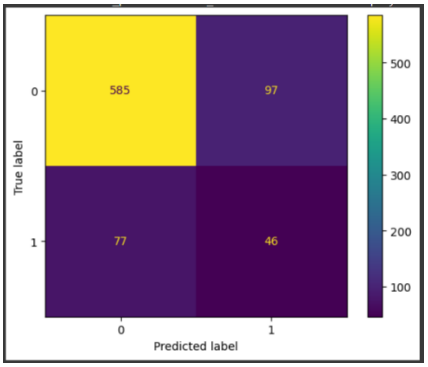
```
#Check updated unique value counts
print(data_frame["Mortality_Status"].value_counts())

Mortality_Status
0    3408
1     616
Name: count, dtype: int64
```

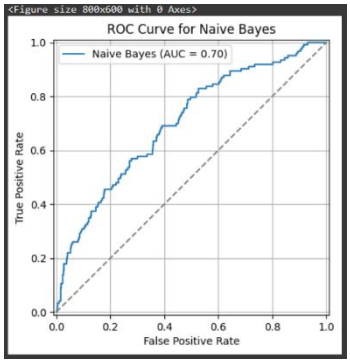
Ensuring all the models are evaluated on the same test set allows for a fair and consistent comparison of model performance (Kuhn & Johnson, 2013). A fixed 'random_state' creates the same data split every time the notebook is executed, allowing results to be reproducible. Additionally, 'stratify=y' preserves the original class proportion of the target variable ("Mortality_Status") in the training and test sets, which is particularly crucial in imbalanced classification issues (Lemaître et al., 2017). If stratification is not used, the model might be trained or tested on unrepresentative class proportions, resulting in biased performance measures.

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

a)

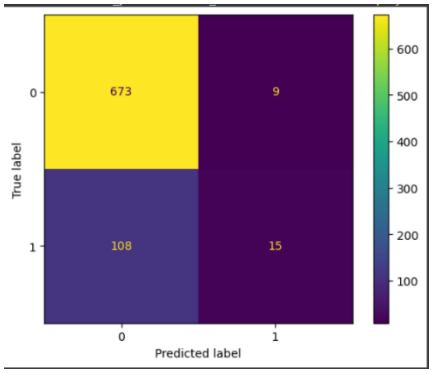
Confusion matrix, the classification report and the AUC-ROC curve graphs for Naïve Bayes

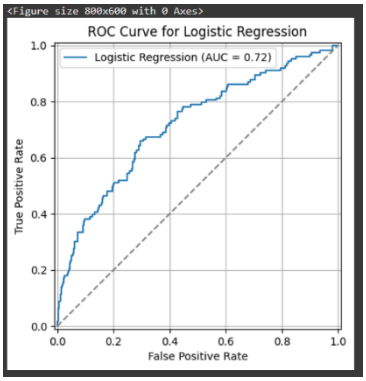





```
Naive Bayes Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.86      0.87       682
           1       0.32      0.37      0.35       123

    accuracy                           0.78       805
   macro avg       0.60      0.62      0.61       805
weighted avg       0.80      0.78      0.79       805

Naive Bayes Accuracy: 0.7838509316770186
Naive Bayes Recall: 0.37398373983739835
Naive Bayes F1 score: 0.3458646616541353
Naive Bayes Precision score: 0.32167832167832167
Naive Bayes ROC-AUC score: 0.615877500417233
Naive Bayes Confusion Matrix:
 [[585  97]
 [ 77  46]]
```

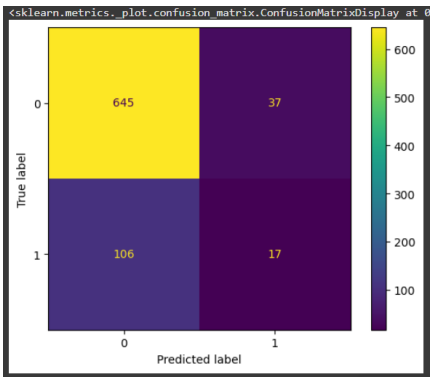
Confusion matrix, the classification report and the AUC-ROC curve graphs for Logistic Regression

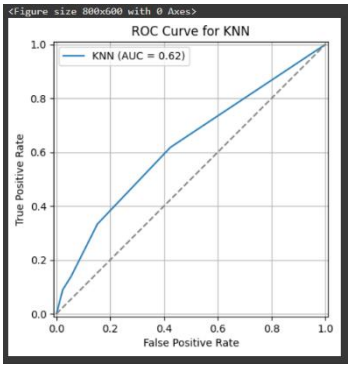





```
Logistic Regression Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.99      0.92       682
           1       0.62      0.12      0.20       123

    accuracy                           0.85       805
   macro avg       0.74      0.55      0.56       805
weighted avg       0.83      0.85      0.81       805

Logistic Regression Accuracy: 0.8546583850931677
Logistic Regression Recall: 0.12195121951219512
Logistic Regression F1 score: 0.2040816326530612
Logistic Regression Precision score: 0.625
Logistic Regression ROC-AUC score: 0.5543773692868894
Logistic Regression Confusion Matrix:
 [[673   9]
 [108  15]]
```

Confusion matrix, the classification report and the AUC-ROC curve graphs for K-Nearest Neighbor






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

           0       0.86      0.95      0.90       682
           1       0.31      0.14      0.19       123

    accuracy                           0.82       805
   macro avg       0.59      0.54      0.55       805
weighted avg       0.78      0.82      0.79       805

KNN Accuracy: 0.822360248447205
KNN Recall: 0.13821138211382114
KNN F1 score: 0.192090395480226
KNN Precision score: 0.3148148148148148
KNN ROC-AUC score: 0.5419795913501657
KNN Confusion Matrix:
 [[645  37]
 [106  17]]
```

b)

Metrices	USE or DO NOT USE	Justification for choosing "USE" or "DO NOT USE" in relation to the success criteria	Model Name	Test Score
Accuracy	USE	More common measurement even though most of the time it is not reliable with unbalanced data	NB	0.78
			LR	0.85
			KNN (K=5)	0.82
Recall	USE	Minimize false negatives	NB	0.37
			LR	0.12
			KNN (K=5)	0.13
Precision	USE	Minimize false positives	NB	0.32
			LR	0.62
			KNN (K=5)	0.19
F-Score	USE	Good for imbalanced datasets	NB	0.35
			LR	0.20
			KNN (K=5)	0.31
AUC-ROC	USE	Display overall model performance	NB	0.61
			LR	0.55
			KNN (K=5)	0.54

c)

Based on the above performance metrics, Naïve Bayes (NB) is the most suitable model for classifying breast cancer mortality status classification. Even though its recall(0.37) is low, it still outperforms Logistic Regression (0.12) and KNN (0.13) in identifying true cases of mortality, which is the ultimate concern in healthcare. It also has the highest AUC-ROC (0.61) and shows best class separability. Even though it's less accurate, its recall and F-Score balance demonstrates it to be more acceptable for few false negatives resulting in identifying high-risk patients.

d)

i)

```
#Defining hyperparameters
from sklearn.model_selection import GridSearchCV
import numpy as np

#Naive Bayes
#Define the parameter grid
param_grid_nb = {
    'var_smoothing': np.logspace(-9, 0, 10)
}

#Instantiate GridSearchCV
nb_gscv = GridSearchCV(GaussianNB(), param_grid_nb, cv=5, scoring='accuracy')

#Fit to the data
nb_gscv.fit(X_train, y_train)
```

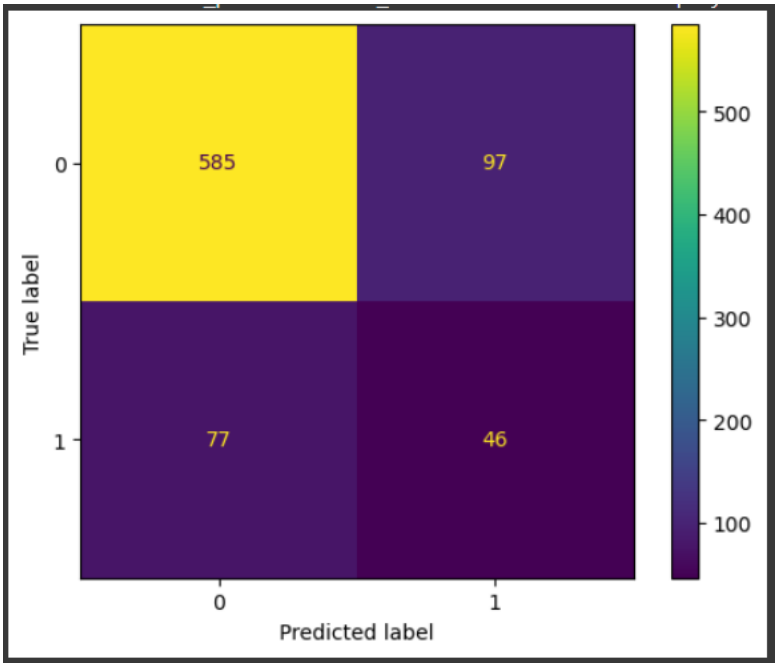
```
GridSearchCV
best_estimator_: GaussianNB
GaussianNB(var_smoothing=np.float64(1.0))
GaussianNB
```

After hyperparameter tuning is done using GridSearchCV, the optimal hyperparameter identified for Naïve Bayes is : var_smoothing : 1.0

This value decreases numerical instability by introducing an infinitesimal value to the variance during the calculation of probability, which improves the model accuracy.

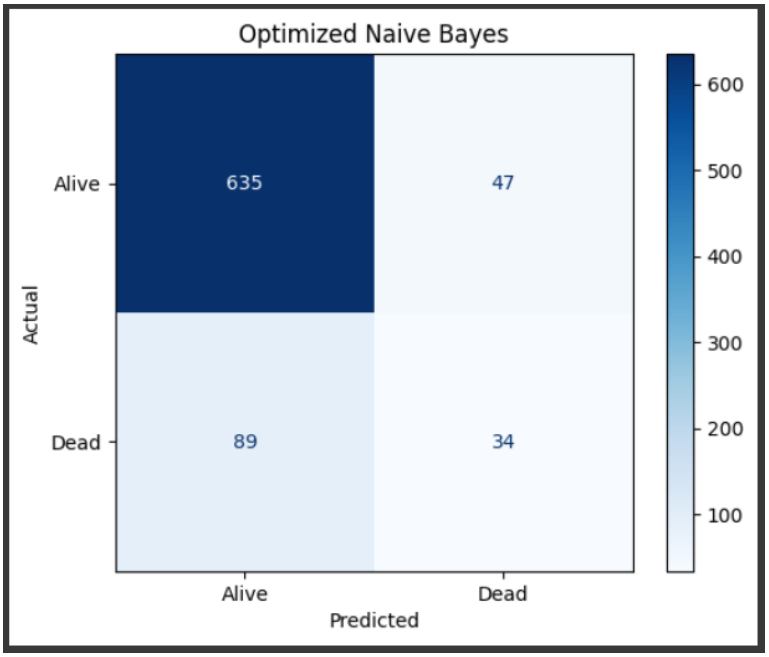
ii)

Before hyperparameter tuning:




```
Naive Bayes Confusion Matrix:
 [[585  97]
 [ 77  46]]
```

After hyperparameter tuning:




```
Naive Bayes Confusion Matrix:
 [[635  47]
 [ 89  34]]
```

After hyperparameter tuning is done, there have been a slight improvement in the positive predictive ability in the Naïve Bayes model. Precision score increased from 0.3217 to 0.4197 after var_smoothing parameter tuning. Even though it's a small increment, this shows a positive improvement in model's ability to precisely identify true positives (correctly predicting the patients' actual death). This aligns well with Task 5)b) by avoiding false positives as it's the most important factor in health context. Furthermore, the F1-score which balance precision and recall in imbalanced datasets also have been slightly increased showing more improvements of the Naïve Bayes classification model performance.

e)

Evaluation of Best-Performing Model (Naïve Bayes)

Even though optimized Naïve Bayes model showed increments in precision score, it still has some limitations:

Limitations:

- **Low Recall for Class 1** – Even after the optimization, the model had only a 0.2764 recall for predicting deceased patients which means it misses a significant amount of mortality cases.
- **Class Imbalance** – The training was done by an imbalanced dataset with extremely higher numbers of alive patients than deceased patients and therefor model is less effective in predicting some mortality cases.
- **Moderate performance** – Only metrics like AUC-ROC (0.603) has a moderate score.

Ethical Issues:

- **Missed Diagnoses and False Diagnoses** – False negative may result in incorrect mortality status prediction.
- **Transparency and Trust** – Patients must be able to understand the model's performance before trusting its output.
- **Data Privacy** – Using a person's sensitive data requires acceptance and permission from strict data protection regulations.

f)

i)

```
#Voting Ensembled Learner
from sklearn.ensemble import VotingClassifier

#Create a dictionary of best 2 models
base_learners = [('KNN', knn_gscv.best_estimator_),('LR', logreg_gscv.best_estimator_)]

#Create voting classifier
ensemble_learner = VotingClassifier(base_learners, voting='soft')
```

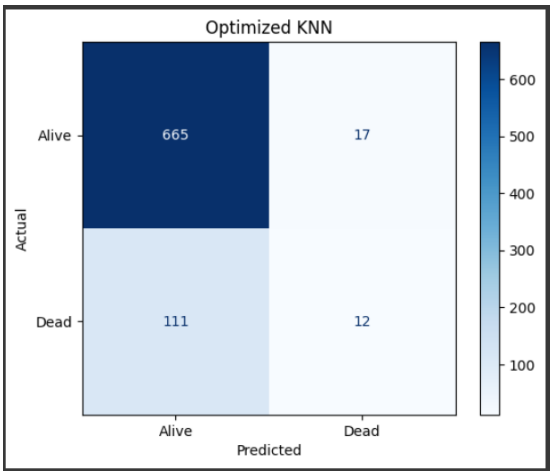
```
[86] #Fit model to training data
     ensemble_learner = ensemble_learner.fit(X_train, y_train)
     y_pred_ensembler = ensemble_learner.predict(X_test)
```

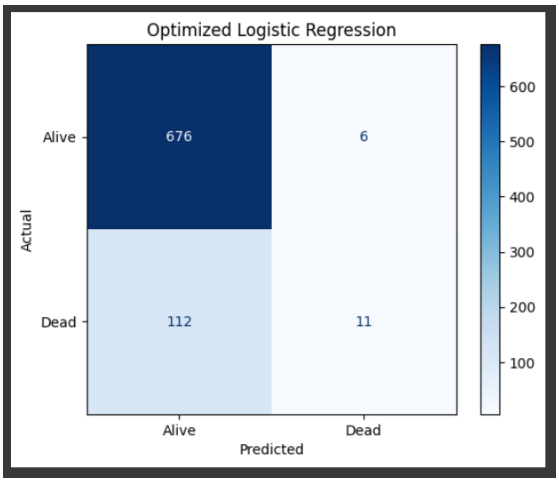
(Importation using a library like joblib was not done and instead a copy of Final Python Notebook 2 was taken and continued to be implemented as the Final Python Notebook 3)

ii)

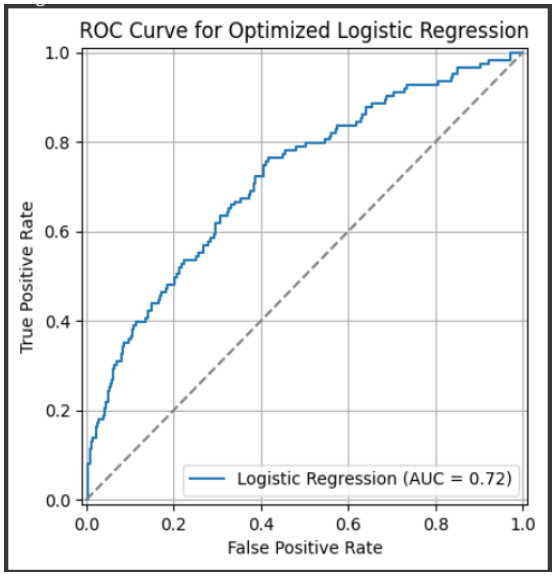
The optimized KNN and Logistic Regression models were used for the ensemble learner.

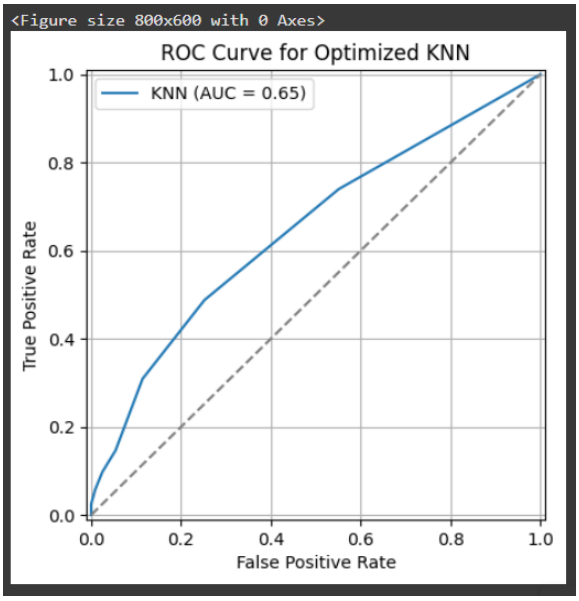
- Confusion matrices of the two models used to create the ensembled learner:





- AUC-ROC Curves of the two models used to create the ensembled learner:





- Classification reports of the two models used to create the ensembled learner:

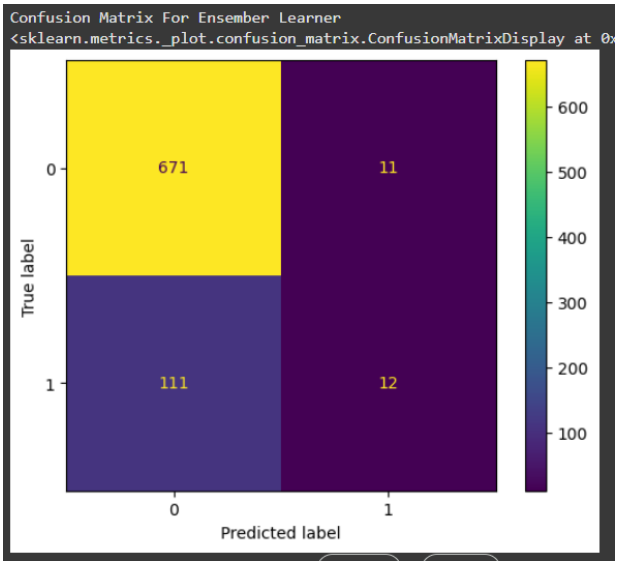
KNN Classification Report:

	precision	recall	f1-score	support
0	0.86	0.98	0.91	682
1	0.41	0.10	0.16	123
accuracy			0.84	805
macro avg	0.64	0.54	0.54	805
weighted avg	0.79	0.84	0.80	805

Logistic Regression Classification Report:

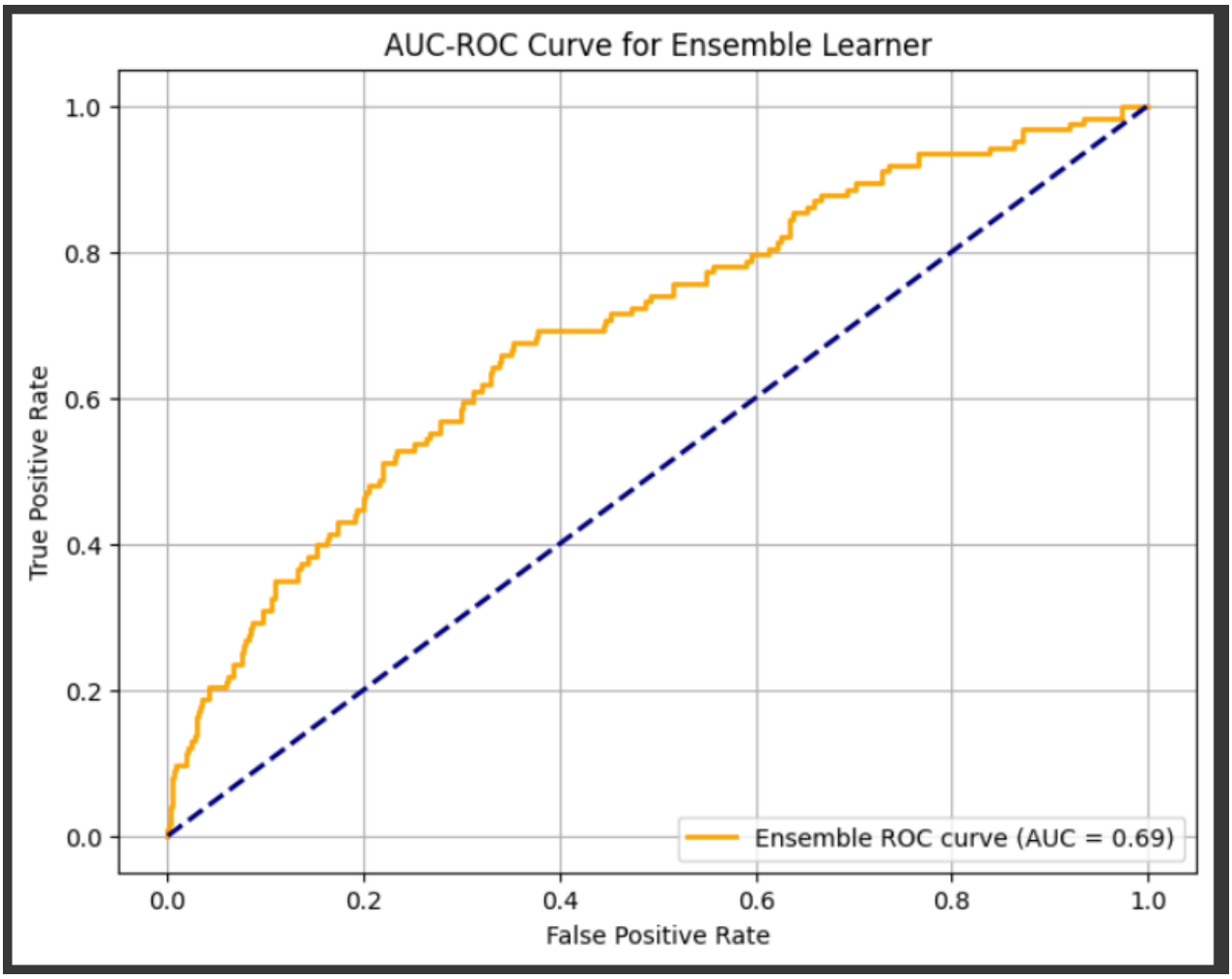
	precision	recall	f1-score	support
0	0.86	0.99	0.92	682
1	0.65	0.09	0.16	123
accuracy			0.85	805
macro avg	0.75	0.54	0.54	805
weighted avg	0.83	0.85	0.80	805

- Confusion matrix, classification report and AUC-ROC curve of Voting Ensemble Classifier:



Classification report for Ensmebler

	precision	recall	f1-score	support
0	0.86	0.98	0.92	682
1	0.52	0.10	0.16	123
accuracy			0.85	805
macro avg	0.69	0.54	0.54	805
weighted avg	0.81	0.85	0.80	805



Justification for using KNN + Logistic Regression

There were multiple reasons for using these models for the ensemble learner:

Strengths :

- Logistic Regression model had the highest accuracy (0.853) and precision (0.647).
- It can perform well in identifying the true positives.
- KNN had a moderate precision (0.41) and as it's a non-parametric learner, it can identify non-linear relationships that Logistic Regression model might miss.

Performance:

- With the combination of Logistic regressor's high precision and KNN's ability to identify non-linear decisions, the Voting Ensemble will achieve an overall well performance.
- LR generalizes trends while KNN brings local sensitivity that balances both weaknesses.

In conclusion, by integrating KNN and LR, the Voting Classifier balances interpretability, performance, and complexity. It is better suited for decision support in clinical settings.

iii)

The Voting Ensemble Learner built by combining KNN and LR models has slightly improved performance that the two base learners when taken individually.

Metric	KNN	LR	Ensemble Learner
Accuracy	0.8409	0.8534	0.8484
Precision	0.4137	0.6470	0.5217
Recall	0.0975	0.0894	0.0975
F1 Score	0.1578	0.1571	0.8017
AUC-ROC	0.5363	0.5403	0.5407

Based on the above improvements, the Ensembled Learner can be recommended to be taken to use instead of individual models for predicting cancer mortality status as it provides more balanced predictions that align with clinical goals.

Case Study (B): Predicting Cancer Patients Survival Months

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

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

```
(4024, 14)
```

Task (2) – Modelling: Build Predictive Regression Models

a)

A Decision Tree Regressor (DTR) has several benefits for survival month modeling in a healthcare prediction problem:

Simple presentation: DTR provides easy and interpretable decision rules, and it is easy for professionals in the medical field to understand how different features affect the survival month.

Non-linearity: DTR can handle complex, non-linear interactions between features which are relevant for interactions that exist in the medical field.

Handling missing values: Decision trees can handle missing values well by using surrogate splits, which are common in medical datasets.

No Feature Scaling is needed: Unlike other models, DTR does not require feature scaling (normalization/ standardization) and makes preprocessing easier.

Due to those above-mentioned advantages, DTR is rather a better option to use when it comes to healthcare prediction problems.

b)

i)

DT-1 :

```
[152] #Regression Decision Trees
      from sklearn.tree import DecisionTreeRegressor
      from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

[153] #Defining targets
      X2 = data_frame2.drop("Survival_Months", axis = 1)
      y2 = data_frame2["Survival_Months"]

      #Spliting data
      X2_train, X2_test, y2_train, y2_test = train_test_split(X2, y2, test_size=0.2, random_state=42

      #Train fully grown decision tree
      full_tree = DecisionTreeRegressor(random_state=42)
      full_tree.fit(X2_train, y2_train)

      DecisionTreeRegressor
      DecisionTreeRegressor(random_state=42)

[155] #Predict
      y2_pred_full = full_tree.predict(X2_test)

      #Evaluate
      mae_full = mean_absolute_error(y2_test, y2_pred_full)
      rmse_full = np.sqrt(mean_squared_error(y2_test, y2_pred_full))
```

DT-2 :

```
[160] #Pruned DT
      #Limiting the tree growth to 4 levels
      pruned_tree = DecisionTreeRegressor(max_depth=4)
      pruned_tree.fit(X2_train, y2_train)

      DecisionTreeRegressor
      DecisionTreeRegressor(max_depth=4)

[161] #Predict
      y_pred_pruned = pruned_tree.predict(X2_test)
```

ii)

In the above given code block, a pruning method has been used for the Decision Tree 2 (DT-2) by limiting the depth to 4 levels by setting the 'max_depth' parameter. This type of pruning is known as **Pre-pruning or early stopping,** where the growth of the decision tree is restricted to prevent the tree being too complex and overfitting the data. There are several advantages and disadvantages to this method:

Advantages-

- **Reduce overfitting**: Since the depth has been made limited, the model will not memorize data and make better predictions on unseen data. This is really an important factor in the healthcare field where overfitting can result in incorrect predictions about new patients.
- **Simple presentation**: A shallow tree is easy to understand by professionals in the medical field.

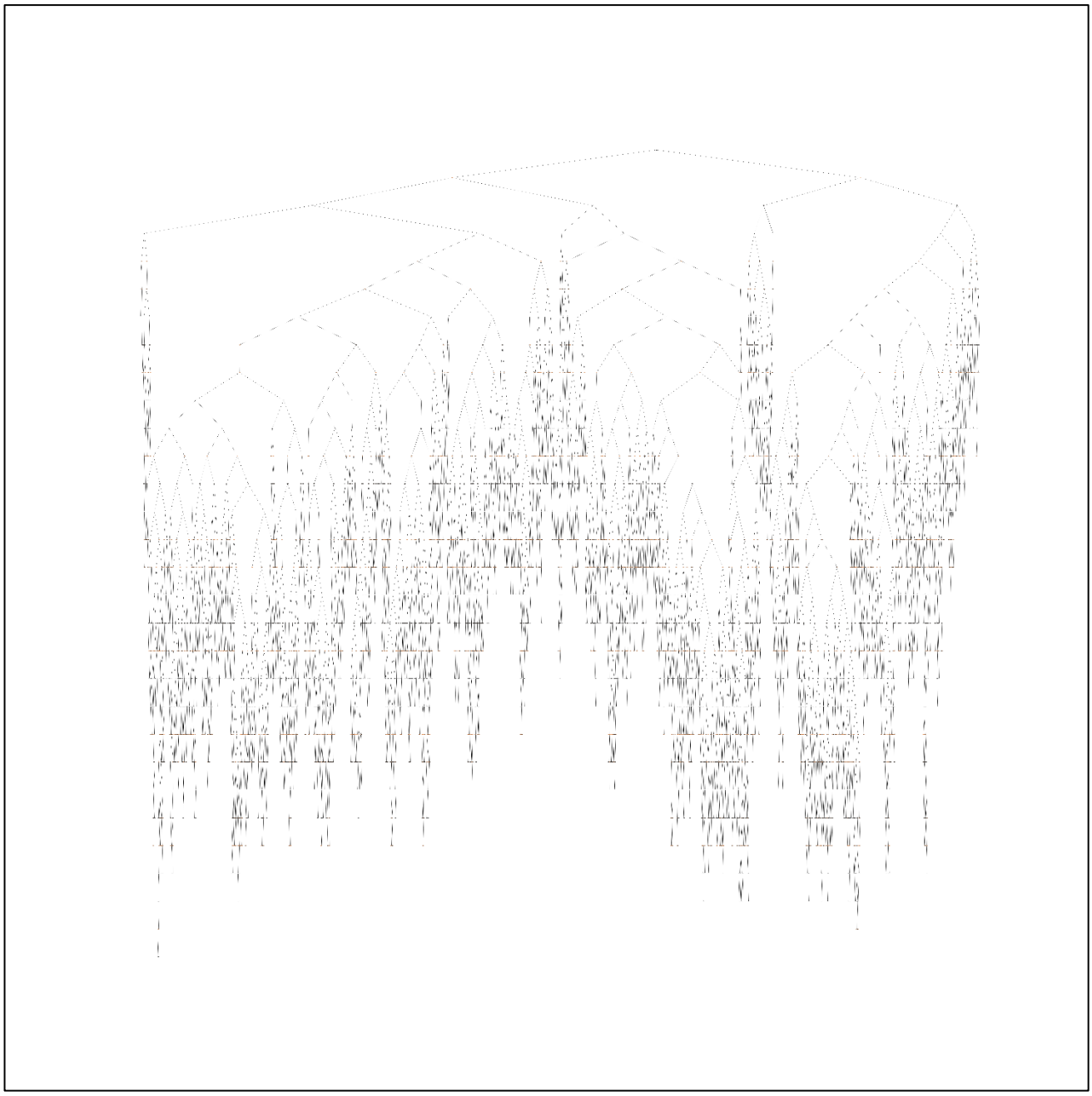
Disadvantages-

- **Underfitting**: If the tree is too shallow, it will not understand the true complexity of the data and will make inaccurate predictions.
- **Loss of details**: If the tree is too shallow, some important relationships between features might be missed.

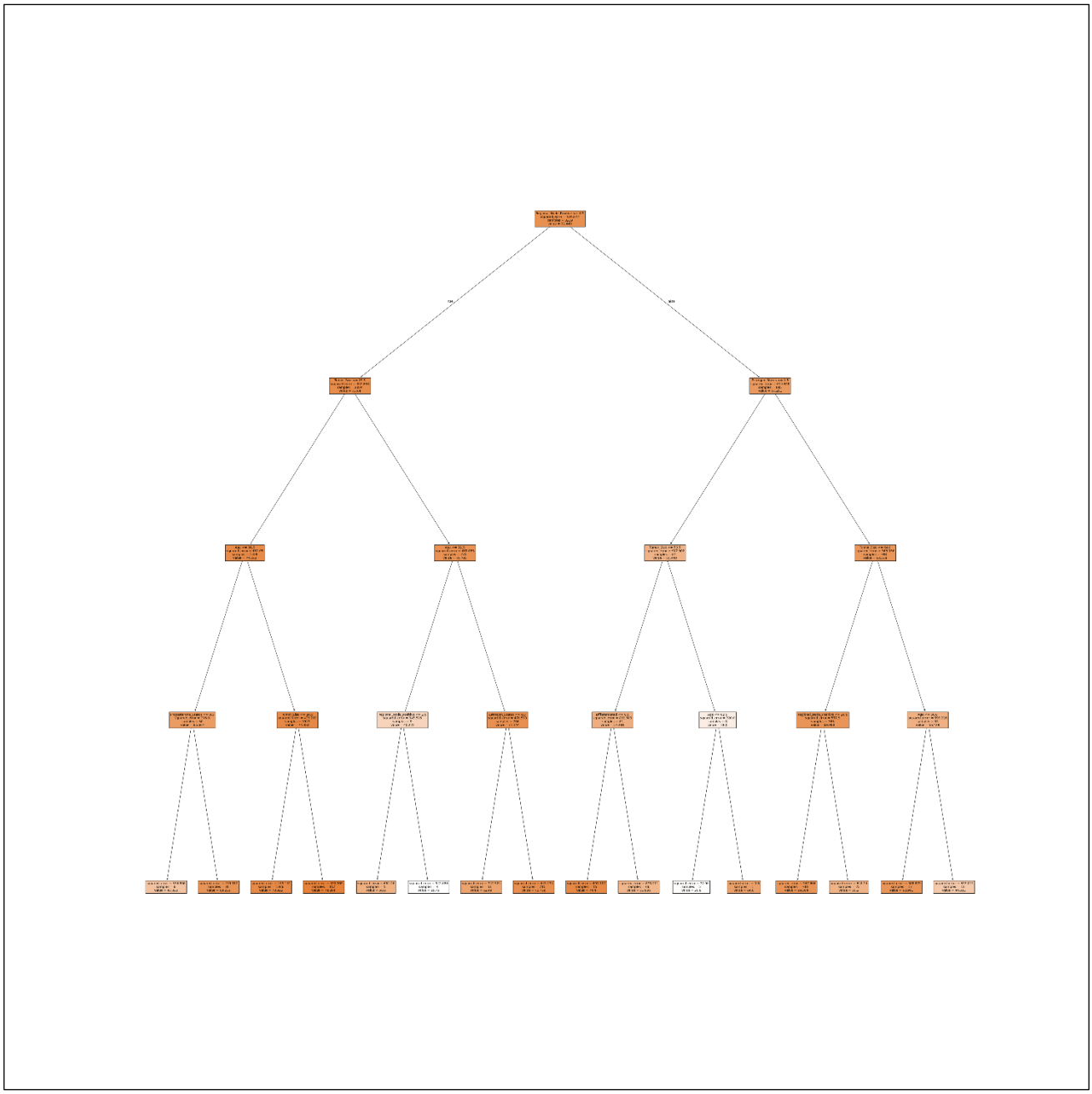
Therefore, the Pre pruning method for decision trees must be done after careful consideration about the dataset and the instance.

c)

DT-1 (Fully grown unpruned tree)



DT-2 (Pruned tree)



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	DO NOT USE	MSE penalizes large error severely. This can corrupt the evaluation of medical data where few large errors can be less harmful than more moderate errors	DT-1	1012.85
			DT-2	489.8575
MAE	USE	MAE gives similar weight to all errors. It's more useful when aligning with healthcare precision and fairness	DT-1	25.3434
			DT-2	18.0811
R-Square	USE	R-Square explains the variance. It helps to understand the model's overall reliability.	DT-1	-1.0042
			DT-2	0.0306

b)

Based on the 'USED' metrics, the best decision tree regression model is **DT-2** (Pruned tree). It has a lower MAE (18.08) compared to DT-1 (25.34), showing that it is more accurate in predicting survival months. Also, DT-2 has a positive R-Squared value meaning it explains the variance of the target. These results indicate that DT-2 has better generalization and makes more accurate, fair predictions more than DT-1 so that it will be more reliable when taking clinical decisions.

c)

While MAE (Mean Absolute Error) and R-Squared were used to evaluate model performance and select the best decision tree (DT-2), there are other important things to keep in check. The MAE gives equal weight to all errors which are desirable in a healthcare environment but might fail to identify the impact of extreme survival month values. However, R-Squared error can be misleading for non-linear models like decision trees when it's a low value. Also, R-Square value does not show how good the model predicts individual values and so, they need to be considered and evaluated clinically to guarantee safety and proper predictions.

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

a)

At first, all the categorical details were converted into numerical data of the patient B002565. Then **DT-2** (Pruned decision tree) was used to estimate the survival months of the patient. It was used to estimate since it has the best performance out of the two decision trees. The model predicted that the patient is expected to survive approximately 56.60 months. The decision path of DT-2 to predict this value is shown below,

1. Regional_Node_Positive <= 4.50 → **Yes** (Patient has 1)
2. Tumor_Size <= 25,50 → **No** (Patient has 41)
3. Age <= 29 → **Yes** (Patient is 29)
4. Regional_Node_Positive <= → **Yes** (Patient has 1)

References

- Crowe, J.P. Jr, Gordon, N.H., Shenk, R.R., Zollinger, R.M. Jr, Brumberg, D.J. & Shuck, J.M. (1992). Primary tumor size: Relevance to breast cancer survival. Archives of Surgery, 127(8), pp.910–915. https://doi.org/10.1001/archsurg.1992.01420080044007.
- Diana, A., Carlino, F., Buono, G., Antoniol, G., Famiglietti, V., De Angelis, C., Carrano, S., Piccolo, A., De Vita, F., Ciardiello, F., Daniele, B., & Arpino, G. (2022). Prognostic relevance of progesterone receptor levels in early luminal-like HER2 negative breast cancer subtypes: A retrospective analysis. Frontiers in Oncology, 12, 813462. https://doi.org/10.3389/fonc.2022.813462
- Giordano, S.H., Cohen, D.S., Buzdar, A.U., Perkins, G. & Hortobagyi, G.N. (2004). Breast carcinoma in men: a population-based study. Cancer, 101(1), pp.51–57. https://doi.org/10.1002/cncr.20312
- Gonzalez, A., Ielpi, G., Romero, A., Gutierrez, M., & Tissera, N. (2007). Breast cancer in young women presents with more aggressive pathologic characteristics: Retrospective analysis from an Argentine national database. The Breast Journal, 13(5), 464–469. https://doi.org/10.1016/j.breast.2007.07.008
- González, A., Ielpi, G., Romero, A., Gutiérrez, M., & Tissera, N. (2020). Breast Cancer in Young Women Presents With More Aggressive Pathologic Characteristics: Retrospective Analysis From an Argentine National Database. JCO Global Oncology, 6, 639–646.
- Kuhn, M. & Johnson, K. (2013). Applied Predictive Modeling. New York: Springer.
- Lemaître, G., Nogueira, F., & Aridas, C.K. (2016). Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. arXiv preprint arXiv:1609.06570. https://doi.org/10.48550/arXiv.1609.06570
- Ohri, N., Haffty, B.G., Buchholz, T.A., Harris, E.E., Woodward, W.A., Shah, C., & Deasy, J.O. (2017). Use of regional nodal irradiation and its association with survival for patients with breast cancer: A National Cancer Database analysis. Advances in Radiation Oncology, 2(4), 403–413. https://doi.org/10.1016/j.adro.2017.07.002
- Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. arXiv preprint arXiv:1811.12808. https://doi.org/10.48550/arXiv.1811.12808
- San Millán-Castillo, R., Morgado, E. & Goya Esteban, R. (2024). On the use of decision tree regression for predicting vibration frequency response of handheld probes. arXiv preprint arXiv:2402.05921. https://doi.org/10.48550/arXiv.2402.05921

- Schwartz, A.M., Henson, D.E., Chen, D., & Rajamarthandan, S. (2014). Histologic grade remains a prognostic factor for breast cancer regardless of the number of positive lymph nodes and tumor size: A study of 161,708 cases of breast cancer from the SEER Program. Archives of Pathology & Laboratory Medicine, 138(8), 1048–1052. https://doi.org/10.5858/arpa.2013-0435-OA
- Yu, K.-D., Cai, Y.-W., Wu, S.-Y., Shui, R.-H., & Shao, Z.-M. (2021). Estrogen receptor-low breast cancer: Biology chaos and treatment paradox. Cancer Communications, 41(10), 968–980. https://doi.org/10.1002/cac2.12191