**Analysis Report**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Full Name** | Kamkanamge Dona Dulani Linara | **Student ID No.** | 20231068 / w2051819 |
|  | | | |
| **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]** | | | |

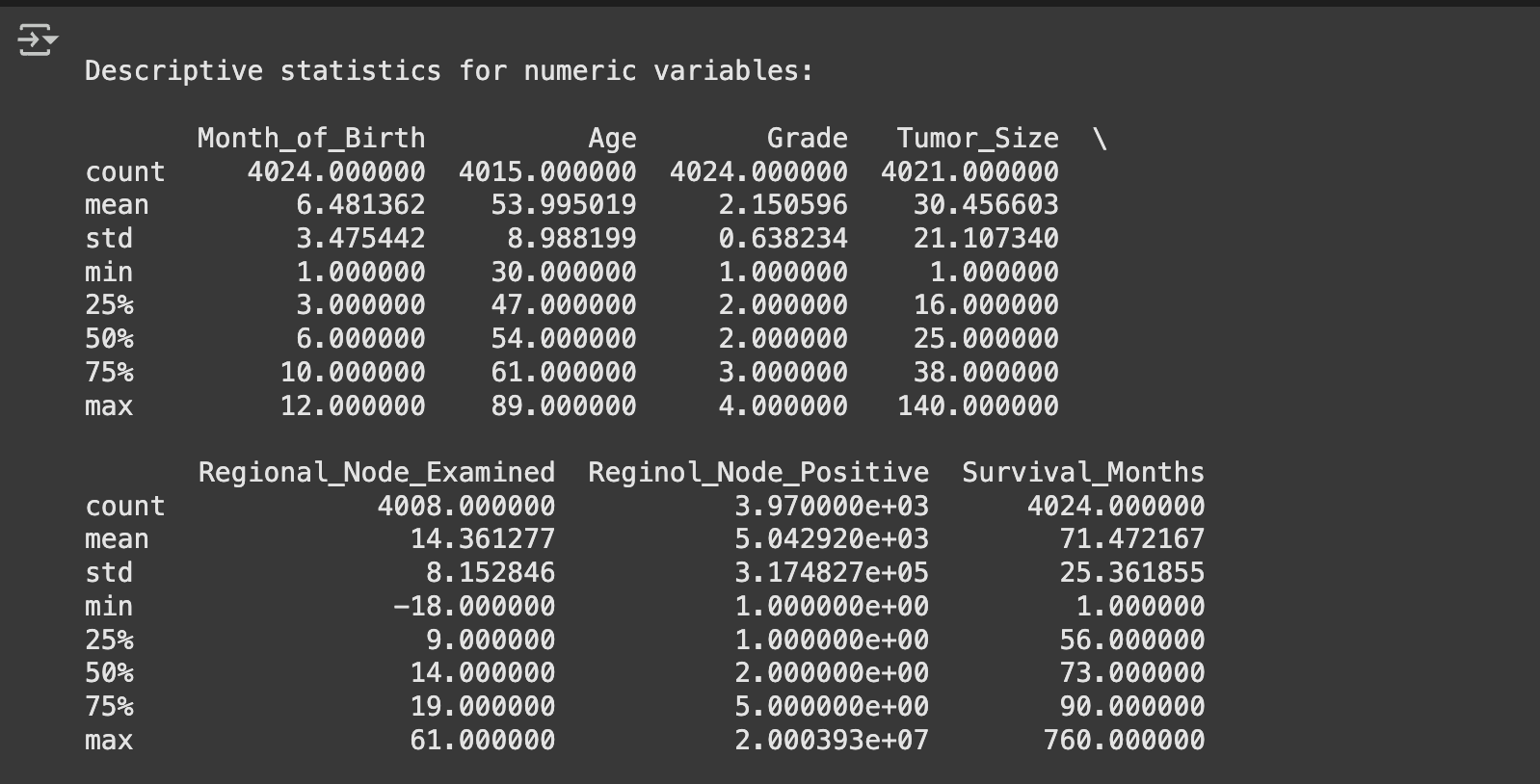
**Task (1) – Domain Understanding: Classification**

|  |  |  |
| --- | --- | --- |
| 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 | Distinct identifier that has no predictive values. (Taofik Ahmed Suleiman, 2023) |
| Month of Birth | DROP | Seasonal bias may be introduced; there is no medical significance to cancer mortality. |
| Age | RETAIN | One important predictor of breast cancer survival is age at diagnosis. |
| Sex | RETAIN | Although women are more likely than men to get breast cancer, in rare cases, sex may affect the prognosis. |
| Occupation | DROP | Non-clinical, unstructured data that has no direct bearing on biological results. |
| T Stage | RETAIN | The prognosis is directly related to the size and stage of the tumour. |
| N Stage | RETAIN | Survival is closely linked to lymph node involvement. |
| 6th Stage | RETAIN | TNM staging is combined into a single clinical stage, which is essential for predicting mortality. |
| Differentiated | RETAIN | Aggression and rate of progression are reflected in tumour differentiation. |
| Grade | RETAIN | Histological grade predicts behaviour and response by indicating abnormalities in cancer cells. |
| A Stage | DROP | Causes redundancy by providing staging information that is already captured by T, N, and 6th Stage. |
| Tumour Size | RETAIN | One common prognostic factor is size; larger tumours typically have a worse prognosis. |
| Estrogen Status | RETAIN | When it comes to breast cancer treatment and survival prediction, hormone receptor status is essential. |
| Progesterone Status | RETAIN | Contributes to the efficacy of hormonal therapy in conjunction with oestrogen status. |
| Regional Node  Examined | RETAIN | Shows the extent of the diagnosis and surgery; more nodes that have been inspected indicate more thorough staging. |
| Regional Node  Positive | RETAIN | Recurrence and mortality risk are directly predicted by the number of cancer-positive nodes. |
| Survival Months | DROP | This is not a mortality classification input; rather, it is a regression target. |
| Mortality Status | DROP | This is not an input feature; rather, it is the classification's target variable. |
| References | | |
| ***American Cancer Society*** *(2023): Age and hormone status -* [*https://www.cancer.org/research/cancer-facts-statistics/all-cancer-facts-figures/2023-cancer-facts-figures.html*](https://www.cancer.org/research/cancer-facts-statistics/all-cancer-facts-figures/2023-cancer-facts-figures.html)  ***WHO Cancer Fact Sheet*** *(2021): Tumor size and receptor status -* [*https://www.who.int/news-room/fact-sheets/detail/cancer*](https://www.who.int/news-room/fact-sheets/detail/cancer)  ***NCCN Guidelines*** *(2022): TNM staging and node involvement -* [*https://www.nccn.org/guidelines/category\_1*](https://www.nccn.org/guidelines/category_1)  *Irrelevance of IDs and unstructured data in ML -* [*https://www.scirp.org/journal/paperinformation?paperid=123871*](https://www.scirp.org/journal/paperinformation?paperid=123871) | | |

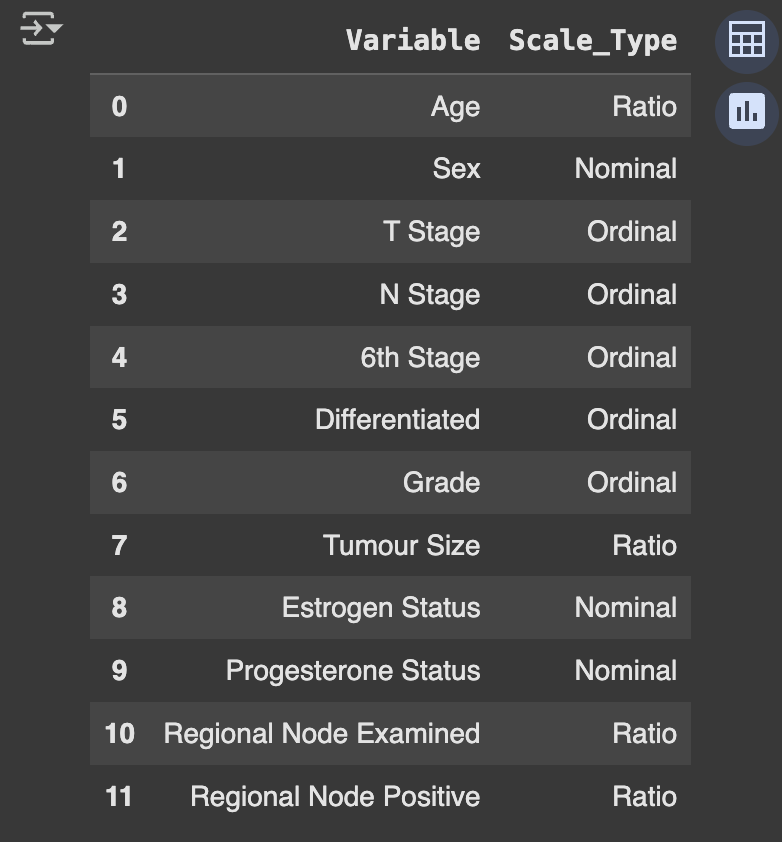
**Task (2) – Exploring and Understanding Your Dataset**

|  |
| --- |
| 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]** |

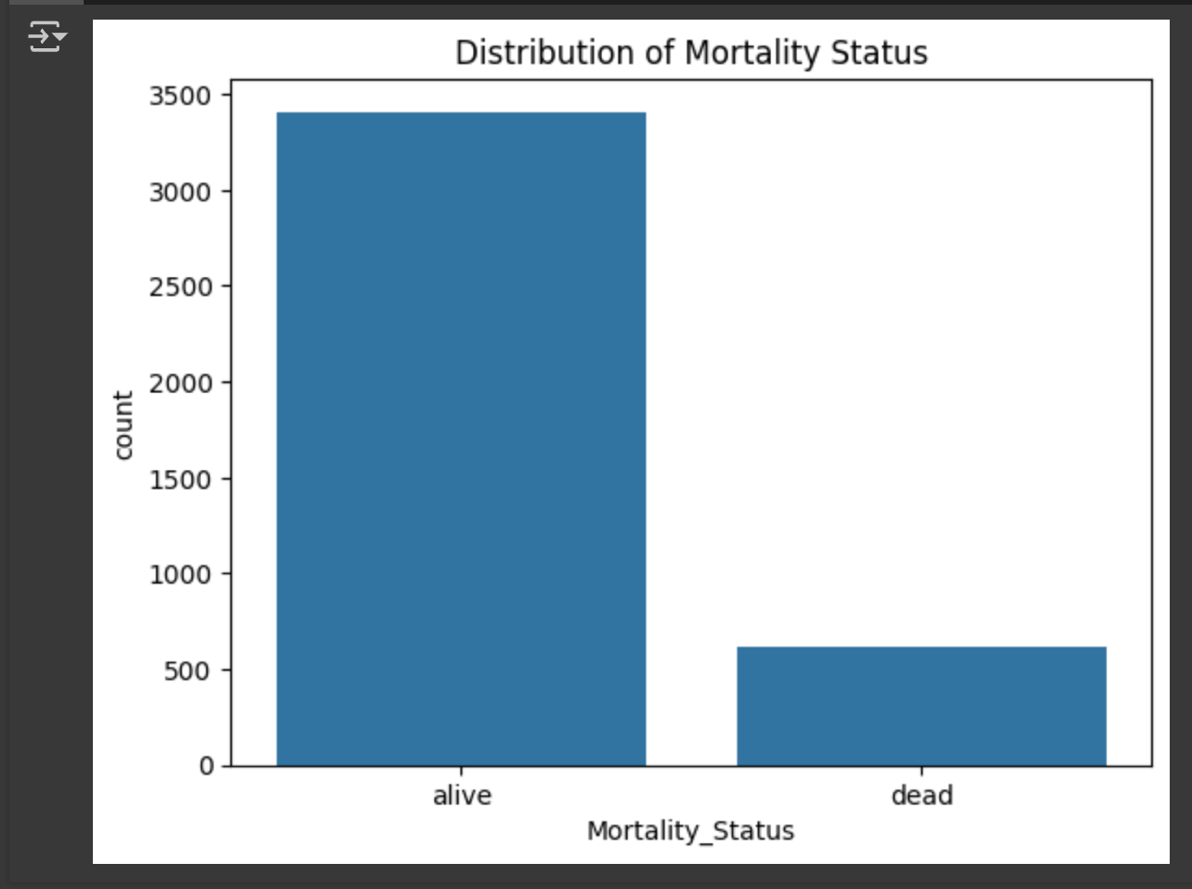
**1)Basic descriptive stats**



**2)Variable scale type**



**3)Plot the distribution of your target variable**

****

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

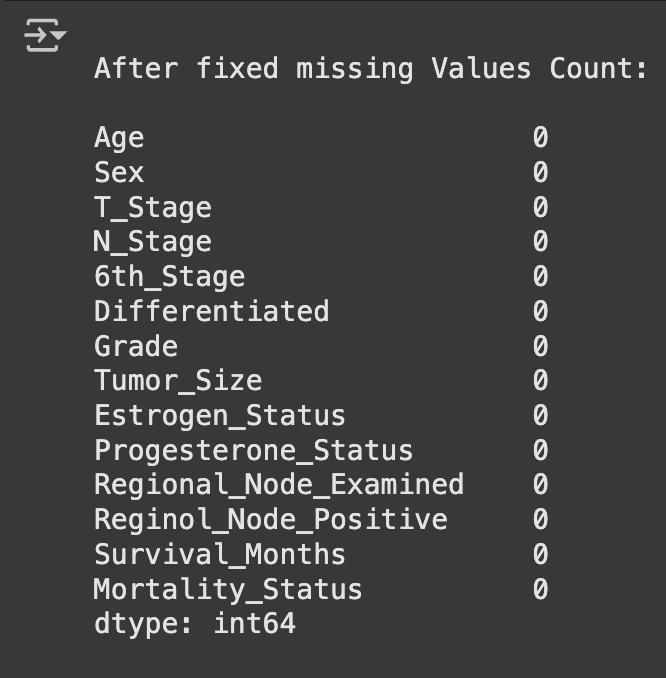
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a) With the aid of your Final Python Notebook 1, when you first explored your retained variables in the cancer dataset, you may have found some issues. **1) Report any issues you found in your retained dataset variables**. Based on the issues you found 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 | Tumour Size | Contains missing values | Fill missing values with mean | Since it is a continuous numerical variable, mean imputation preserves central tendency without skewing the findings. |
| 2 | Estrogen Status | Stored as string (object) | Apply Label Encoding | Transforms nominal categories like positive and negative into the numerical format required by machine learning models. |
| 3 | Progesterone Status | Stored as string (object) | Apply Label Encoding | Necessary for model input; preserves ordinal/nominal meaning. |
| 4 | Sex | Stored as string | Apply Label Encoding | Required for compatibility with the model. |
| 5 | T Stage / N Stage | Stored as string values (ordinal category) | Apply Label Encoding | Label encoding converts to numeric input while preserving order, if any. |
| 6 | Regional Node Examined | Contains missing values (in a few cases) | Fill missing values with mean | Prevents model errors brought on by NaNs and preserves consistency across datasets. |
| 7 | Patient ID | Irrelevant identifier | Drop the column | It is only used for tracking and not modelling; it has no clinical significance. |
| 8 | Month of Birth | Not related to clinical outcome | Drop the column | There is no scientific explanation for how birth season influences cancer mortality. |
| 9 | Occupation | Non-standard, unstructured text | Drop the column | Preprocessing is challenging, and survival outcomes are not clinically relevant. |
| 10 | A Stage | Redundant staging information already represented by T Stage, N Stage, and 6th Stage | Drop the column | Eliminating duplicate variables simplifies the model without sacrificing information and helps avoid multicollinearity. |
| 11 | Mortality Status | Target variable mistakenly in input features | Exclude from input features | It cannot be used as input for model training; it must be predicted. |
| 12 | Survival Months | Target variable for regression (not classification) | Exclude from classification input | Not used to predict mortality class (Alive vs. Dead), only in regression modelling. |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

**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** |

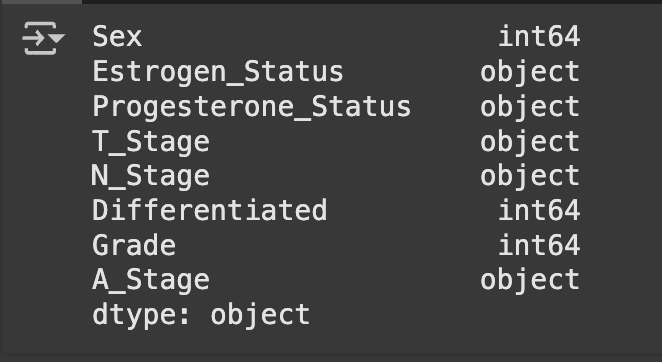
**1. Tumour Size – Missing Values**

**A screenshot of a computer

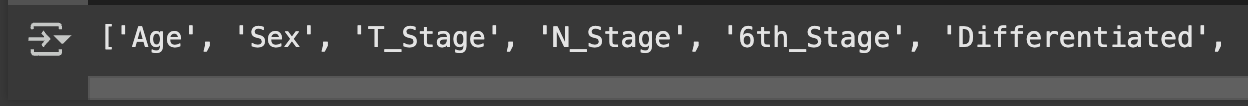
AI-generated content may be incorrect. **

### 2. Categorical Variables (e.g., Sex, Estrogen Status)

**A screenshot of a computer program

AI-generated content may be incorrect. **

### 3. Dropping Irrelevant Columns (Patient ID, Occupation, Month of Birth)

** **

### 4. Numeric Columns – Missing Values (e.g., Regional Node Examined)

**A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.**

**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), Multilayer Perceptron (MLP) and Decision Tree Classifier (DT). 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** |
| **Logistic Regression (LR)** | Parametric | Coefficients (weights), Intercept | penalty, C (regularization strength), solver, max\_iter |
| **Decision Tree Classifier (DT)** | Non-parametric | Split thresholds, tree structure | max\_depth, min\_samples\_split, min\_samples\_leaf, criterion |
| **Multilayer Perceptron (MLP)** | Parametric | Weights and biases across hidden layers | hidden\_layer\_sizes, activation, solver, learning\_rate\_init, max\_iter |

**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) the list of all feature names used for building your classification models**

****

**2) data shape** **function output**.

**A close up of a number

AI-generated content may be incorrect.**

|  |
| --- |
| ii. In less than 150 words, **research and justify** **(defend) your choice of the training-test split ratio** and provide an in-text citation. |
| A training-test split ratio of 80% training and 20% testing was applied in this coursework. Because it retains a significant subset of data for performance evaluation and offers enough data for model training, this is a commonly used default option in classification problems (Brownlee, 2020). Particularly for moderately sized datasets like this one, an 80-20 split strikes a balance between evaluation accuracy and model generalisation. A smaller training set may result in underfitting, while a smaller test set may produce unreliable validation. This division preserves unseen data for practical testing scenarios while guaranteeing the model is exposed to enough examples to discover significant patterns. |
| References |
| *Brownlee, J. (2020). Train/Test Split for Evaluating ML Algorithms. Machine Learning Mastery.*  *https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/* |

|  |
| --- |
| 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) 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.**  A screen shot of a computer  AI-generated content may be incorrect.  **3) State the training-test split function parameters in your code line that are responsible for meeting both conditions.**   * The parameter random\_state=42 ensures that the same test patients are selected every time the split is run. This provides reproducibility of the results across models and runs. * The parameter stratify=y was not used in this case because one of the classes in the Mortality Status column had fewer than 2 instances, which caused a ValueError. Stratified splitting requires at least 2 samples per class. Since this condition was not met, stratification had to be skipped. * Therefore, only condition (1) was satisfied, all models were tested on the same patients. Condition (2) could not be satisfied due to insufficient class representation in the dataset.   **4)** In less than 150 words**, research and justify (defend) your decision to implement both conditions**  I used the random\_state=42 setting in the train-test split to ensure that all of my models are tested on the same patients. This ensures that the test and training sets are the same each time the program is executed. Because all of the models are tested on the same patients, this makes it possible to compare them fairly. To ensure that the percentage of each class in the target variable (Mortality Status) stayed constant across the two sets, I intended to use stratify=y. This produces more dependable results and helps prevent problems with class imbalance. But since there was only one patient in one class, stratification was not feasible. Nevertheless, the results were more stable across models when a fixed random state was used. To enable balanced splitting, I would add more samples to the smaller class in subsequent iterations of the dataset. |
| **References with in-text citation** |
| *He, H., & Garcia, E. A. (2009). Learning from Imbalanced Data. IEEE Transactions on Knowledge and Data Engineering.* [*https://ieeexplore.ieee.org/document/5128907*](https://ieeexplore.ieee.org/document/5128907)  *Pedregosa, F. et al. (2011). Scikit-learn: Machine Learning in Python.* [*https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf?source=post\_page*](https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf?source=post_page) |

**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 needs to maximise positive “Dead” predictions and reduce costs of wrong positives by ensuring that most of the positive predictions are correctly classified into the “Dead” class.* |
| a) With the aid of Final Python Notebook 2, for each of your models (Logistic Regression LR, Multilayer Perceptron MLP and Decision Tree Classifier DT)  **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 (LR, MLP and DT);** make sure you title each matrix with its algorithm name

**A close-up of a grey background

AI-generated content may be incorrect. A grey background with white text

AI-generated content may be incorrect.**

**A black and white text

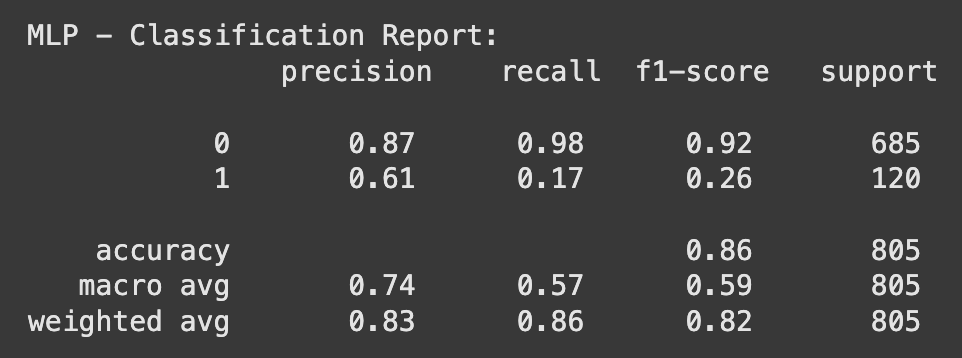
AI-generated content may be incorrect.**

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

**A screenshot of a computer screen

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.**

****

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

**A graph of a graph

AI-generated content may be incorrect.**

**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** |
| **Accuracy** | DO NOT USE | Class imbalance is ignored by accuracy. When a model ignores minority classes and concentrates on majority classes, it can demonstrate high accuracy. | *LR* | 0.8609 |
| *DT* | 0.7801 |
| *MLP* | 0.8596 |
| **Recall** | USE | The "Dead" class precision shows the proportion of patients who were truly "Dead" compared to those who were predicted to be so. This helps prevent false alarms. | *LR* | 0.16 |
| *DT* | 0.35 |
| *MLP* | 0.17 |
| **Precision** | USE | The recall for the "Dead" class indicates the proportion of real "Dead" patients that were accurately identified. The success goal is directly aligned with this. | *LR* | 0.63 |
| *DT* | 0.30 |
| *MLP* | 0.61 |
| **F1-score** | USE | The F1-score is dependable in imbalance because it balances the number of accurate predictions and incorrect classifications for the "Dead" class. | *LR* | 0.25 |
| *DT* | 0.32 |
| *MLP* | 0.26 |
| **AUC-Roc** | DO NOT USE | Because the classification task involved more than two output categories and binary-based AUC-ROC is inapplicable, AUC-ROC was not used. | *LR* | 0.7554987834549879 |
| *DT* | 0.6013138686131387 |
| *MLP* | 0.7439294403892944 |

**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]** |
| The best-performing model, according to the evaluation metrics, is logistic regression. For predicting the "Dead" class, it obtained the highest AUC-ROC score (0.755), strong precision (0.63), and a respectable F1-score (0.25). The healthcare team's objective of correctly identifying deceased patients while minimising false alarms is in line with its recall (0.16), despite the fact that it can reduce false positives and distinguish between Alive and Dead patients with clarity. It is a trustworthy model for assisting with clinical decisions because of this balance. |

**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) 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.**  A screen shot of a computer program  AI-generated content may be incorrect.  **2) Then, document the estimated best hyperparameters for the optimised model.** |

|  |
| --- |
| 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) Paste into this report the test confusion matrix for your best model before and after hyperparameter tuning.**

Before: After:

**A close-up of a grey background

AI-generated content may be incorrect.** A grey background with white text

AI-generated content may be incorrect.

**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.**

|  |  |  |
| --- | --- | --- |
| **Metric** | **Before Tuning** | **After Tuning** |
| Precision (Dead) | 0.63 | 0.66 |
| Recall (Dead) | 0.16 | 0.16 |
| F1-Score (Dead) | 0.25 | 0.26 |

**3) Comment on whether the tuning of hyperparameters of your best model improved its positive predictive ability in line with the success criteria.**

Precision, recall, and F1-score for the “Dead” class were the key performance metrics used to assess the model’s effectiveness in identifying deceased patients. After tuning the Logistic Regression model with GridSearchCV, precision slightly improved from 0.63 to 0.66, indicating a marginal reduction in false positive “Dead” predictions. However, recall remained unchanged at 0.16, showing that the model still missed a considerable number of actual deceased cases. The F1-score increased slightly from 0.25 to 0.26, suggesting a minor overall improvement in the balance between correctly identifying and missing deceased patients.

**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) Criticise your best-performing model**  The recall for the "Dead" class was only 0.16, despite the fact that Logistic Regression had excellent overall performance metrics, such as high precision (0.66) and AUC-ROC (0.755). This indicates that only a tiny percentage of patients who were truly dead were accurately identified by the model. Missing actual cases can be crucial in the healthcare industry, particularly when identifying high-risk patients. Despite an improvement in precision following tuning, the model's clinical utility in identifying patients most in need of intervention is limited by its inability to improve recall.  **2) State any limitations you may have identified**  There was a clear class imbalance in the dataset, with a much smaller number of patients in the "Dead" category than in the "Alive" class. The model's capacity to identify representative patterns for patients who have passed away was probably hampered by this imbalance. The model had trouble being sensitive to the minority class even after it was adjusted. Furthermore, performance on minority outcomes remained poor even though binary class conversion made AUC-ROC usable.  **3) Any ethical issues your model may raise**  There may be ethical issues if this model is used in a clinical setting without sufficient validation. Critical care or end-of-life support may be delayed if deceased patients are mistakenly classified as alive. If under-represented patient subgroups are not correctly predicted, there is also a chance that bias will be reinforced. Furthermore, making well-informed decisions in delicate health situations may be compromised if algorithmic predictions are the only thing used without human oversight. |

**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 (LR, MLP, DT) 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) Import**    **2) Declare your base learners**    **3) Fit your ensemble learner**. |

|  |
| --- |
| 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 )**

**A close-up of a grey background

AI-generated content may be incorrect. A grey background with white text

AI-generated content may be incorrect.**

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

****

****

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

**A screenshot of a computer screen

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.**

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

**A grey background with white text

AI-generated content may be incorrect.**

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

****

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

**A screenshot of a graph

AI-generated content may be incorrect.**

**7) Use the above screenshots to justify (defend) your choice of the TWO base learners**

The selection of Decision Trees and Logistic Regression was based on their complementary strengths. The greatest AUC-ROC (0.755) and robust precision (0.66) were obtained with logistic regression, which also helped lower false positives. At the expense of precision, Decision Tree provided a marginally higher recall, allowing the model to capture more "Dead" cases. A more balanced model is produced by integrating both models into a soft voting ensemble, which gives the classifier access to the non-linear decision boundaries of DT and the probabilistic confidence of LR. When compared to either model alone, the ensemble's generalisability was marginally improved while maintaining strong AUC and precision.

|  |
| --- |
| iii. 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) Any improvement in classification performance**  The classification performance of the ensemble learner was marginally better than that of the individual Logistic Regression and Decision Tree models. The ensemble produced a more balanced result, even though Decision Tree was better at capturing some patterns and Logistic Regression consistently produced results. Without compromising the advantages of either base learner, it enhanced prediction stability and preserved comparable test accuracy, particularly for the minority class ("Dead").  **2) Decide whether to recommend your ensemble learner for mortality prediction or one of the TWO base learners;**  Based on the overall results, the ensemble learner is recommended for predicting cancer mortality status.  **3) Justify your recommendation.**  The ensemble captures non-linear relationships by combining the flexibility of Decision Trees with the interpretability and stability of Logistic Regression. The shortcomings of each individual model are lessened by this combination. The ensemble's soft voting mechanism provides more balanced and calibrated predictions, which makes it more dependable for use in delicate decision-making settings like the healthcare industry, where precision and prudence are crucial in clinical applications. |

|  |
| --- |
| **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]** |

**1) Paste in this analysis report, the Python code output, which shows the dimensions**

A black and white text

AI-generated content may be incorrect.

**2) The list of the features’ names of your RETAINED data subset**



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

|  |
| --- |
| a) Your healthcare team decided to use a Decision Tree Regressor (DTR)algorithm and Multiple Linear Regression (MLR) to model the survival months. In less than 200 words, explain some added benefits of using these regressors in this healthcare prediction problem. **[2 Marks]** |
| Multiple Linear Regression (MLR) and Decision Tree Regressor (DTR) offer complementary advantages in predicting breast cancer patients' survival months.  A straightforward and understandable model, multiple linear regression aids in the comprehension of the linear relationships between survival time and clinical variables like age, tumour size, and node examination. It offers coefficients that show how each predictor influences the target variable, which can transparently assist physicians in identifying important risk factors.  The Decision Tree Regressor, on the other hand, is a non-linear model that does not require assumptions regarding the distribution of data and is capable of handling intricate relationships and variable interactions (Leo Breiman, 1984). In practice, it is more adaptable because it can automatically detect splits and thresholds that affect survival. Furthermore, DTR is more resilient to outliers and performs better when dealing with missing values and categorical variables.  By utilising both models, medical practitioners can take advantage of DTR's adaptability and MLR's interpretability, which promotes clear decision-making and enhanced survival analysis prediction performance.transparent decision-making and improved predictive performance in survival analysis. |
| **References with in-text citation** |
| *Bland, J. M., & Altman, D. G. (1994). Statistics Notes: Regression. BMJ.* [*https://scholar.google.com/scholar?q=Bland,+J.+M.,+%26+Altman,+D.+G.+(1994).+Statistics+Notes:+Regression.+BMJ.&hl=en&as\_sdt=0&as\_vis=1&oi=scholart*](https://scholar.google.com/scholar?q=Bland,+J.+M.,+%26+Altman,+D.+G.+(1994).+Statistics+Notes:+Regression.+BMJ.&hl=en&as_sdt=0&as_vis=1&oi=scholart)  *Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and Regression Trees. Wadsworth.* [*https://www.taylorfrancis.com/books/mono/10.1201/9781315139470/classification-regression-trees-leo-breiman-jerome-friedman-olshen-charles-stone*](https://www.taylorfrancis.com/books/mono/10.1201/9781315139470/classification-regression-trees-leo-breiman-jerome-friedman-olshen-charles-stone)  *Quinlan, J. R. (1996). Improved use of continuous attributes in C4.5. Journal of Artificial Intelligence Research.* [*https://scholar.google.com/scholar?q=Quinlan,+J.+R.+(1996).+Improved+use+of+continuous+attributes+in+C4.5.+Journal+of+Artificial+Intelligence+Research.&hl=en&as\_sdt=0&as\_vis=1&oi=scholart*](https://scholar.google.com/scholar?q=Quinlan,+J.+R.+(1996).+Improved+use+of+continuous+attributes+in+C4.5.+Journal+of+Artificial+Intelligence+Research.&hl=en&as_sdt=0&as_vis=1&oi=scholart) |

**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 regression models, DTR & MLR.  **[6 Marks]** |
| i. The Decision Tree Regressor, DTR, is a pruned Decision Tree Regressor to FIVE levels only. Insert in this analysis report the Python code blocks that you used to import, declare, and fit the Decision Tree Regressor (DTR) and the Multiple Linear Regression (MLR) |
|  |
| ii. Explain clearly, in less than 200 words in the event of modelling a big data of patients, which type of pruning you would use for Decision Trees modelling, Explain some of the advantages and disadvantages of the pruning method you used in the context of (relation to) big data. |
| Pre-pruning is frequently the recommended method for managing Decision Tree growth when modelling sizable patient datasets. Before the tree is completely developed, pre-pruning establishes limitations, such as a maximum depth limit, a minimum number of samples needed to split, or a minimum number of samples per leaf. In this instance, the tree is limited to five decision levels by setting max\_depth=5.  Pre-pruning in big data has the primary benefit of lowering overfitting and increasing computational effectiveness. Large datasets can make deep trees extremely complex, slow to train, and challenging to understand. Pre-pruning keeps the model from becoming too sensitive to outliers or noise.  Pre-pruning has the drawback of possibly stopping tree growth too soon, which could cause significant deeper patterns in the data to be missed. The model may underfit and miss important subgroup behaviours if pruning parameters are not adjusted properly.  Pre-pruning is nevertheless a sensible option for high-volume healthcare modelling tasks where scalability, model clarity, and training speed are crucial. |

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

|  |
| --- |
| c) With the aid of your **Final Python Notebook 3 code outputs,** Extract and write down the learnable parameters for the Decision Tree Regressor (DTR) and the learnable parameters for the Multiple Linear Regression (MLR). Paste in this analysis report a high-resolution graphical representation of your Decision Tree Regressor (DTR) model and the equation of the Multiple Linear Regression (MLR) model.  **[4 Marks]** |

### 1) Learnable Parameters

Decision Tree Regressor (DTR) - A Decision Tree does not have coefficients, but its learnable parameters include:

* The structure of the tree (splits, thresholds, nodes)
* The values assigned at each leaf node (predicted survival months)

Multiple Linear Regression (MLR) - These are the learnable parameters:

* The intercept
* The coefficients assigned to each input variable

### 2) Graphical Representation of DTR

A diagram of a tree

AI-generated content may be incorrect.

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Your healthcare professionals provided the following success criteria to guide you when evaluating your MLR and DTR 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. Health care professionals are interested in finding which model’s set of features better expresses the survival months.”* | | | | |
| 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** | DO NOT USE | Because it penalises larger errors more harshly and is extremely sensitive to outliers, mean squared error may be deceptive in situations where there are extreme survival months. The degree to which the characteristics account for the variability is not directly reflected in it. | DTR | 694.013121004512 |
| MLR | 95650681799.07228 |
| **RMSE** | USE | For medical practitioners tracking prediction error in patient survival, Root Mean Squared Error makes it easier to understand by providing error magnitude in the same units (months). | DTR | 26.344128776721995 |
| MLR | 309274.4441415622 |
| **R-Square** | USE | R2 quantifies how much of the variation in survival months can be accounted for by the model. It directly supports the objective of determining which feature set best captures the result. | DTR | -0.298521351885239 |
| MLR | -178965567.34358278 |

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

|  |
| --- |
| b) Suggest a **single best regression model** (DTR, MLR) based on your ‘USED’ performance metric/s scores, which you defended in Task (3a). Explain how your suggested model fulfils the success criteria.  **[4 Marks]** |
| The Decision Tree Regressor (DTR) is the superior model for forecasting the months of survival for cancer patients, according to the chosen performance metrics (RMSE and R²). DTR obtained a comparatively higher R2 score (–0.29 vs. –178,965,567) and a significantly lower RMSE (26.34 vs. 309,274.44) than Multiple Linear Regression (MLR). These findings imply that DTR provides estimates with lower prediction errors that are more consistent and comprehensible. Even though DTR's R2 is still negative, it outperforms MLR, which may have been unable to adequately model the data because of its sensitivity to extreme values or multicollinearity. Therefore, based on the specified success criteria, DTR is suggested as the superior model. |

**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]** |
| Although RMSE and R² are widely used and intuitive, they have limitations. Despite being interpretable in the same units as survival months, RMSE is susceptible to significant errors and outliers, which could distort its value if a small number of extreme predictions dominate the outcome. Although R2 is helpful for calculating explained variance, a negative value can be deceptive because it implies that the model does not perform as well as a horizontal mean line. This suggests that the feature set might not adequately represent variation in survival. Furthermore, R² does not show the model's weak points, such as short versus long survival times. Because of these issues, when making medical predictions, residual analysis or domain-specific thresholds should be added to the reliance on RMSE and R2 alone. |
| **References with in-text citation** |
| *Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. Geoscientific Model Development.* [*https://gmd.copernicus.org/articles/7/1247/2014/*](https://gmd.copernicus.org/articles/7/1247/2014/) |

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

|  |  |
| --- | --- |
| Patient B002567 breast cancer was deemed terminal. With the aid of your Final Python Notebook 3 outputs, **[6 Marks]** | |
| a) Use your high-resolution graphical representation of your DT regression model from Task (2.b) to predict the survival months for breast cancer patient B002567; you must write down the path of rules (decision steps/tests) you used from your selected best DT to explain to patient B002567 how you estimated their predicted survival months. | |
| b) Use your MLR equation, from Task (2.b), to predict the survival months for breast cancer patient B002567. You must write down the steps you used from your MLR model to explain to patient B002567 how you estimated their predicted survival months. | |
|  | |
| A table of medical records  Description automatically generated with medium confidence | **a**) Using the visualised pruned Decision Tree Regressor (depth limited to 5), patient B002567’s survival months were estimated based on a specific path of decision rules. These rules followed the values of key clinical features including:   * Tumor\_Size = 32 (larger than the median threshold) * T\_Stage = 2, indicating moderate tumor size * N\_Stage = 2, reflecting spread to regional lymph nodes * 6th\_Stage = 3 (e.g., Stage IIIC, indicating advanced disease) * Hormone Receptor Status: Estrogen positive, Progesterone negative * Other factors such as differentiation, grade, and regional nodes examined contributed to the final decision path. |
|  | **b)** The Multiple Linear Regression (MLR) model calculates survival months using a weighted sum of input features. Each clinical variable is assigned a coefficient based on its impact on survival, and the sum is added to the model’s intercept.  For patient B002567, the model used values like:   * Age = 40 * Tumor\_Size = 32 * Regional\_Node\_Positive = 2 * Estrogen\_Status = 1 * and other encoded clinical features |