**Homework 3**

Group 4

Dennis Jose

Vamsinath Thatha

jose.de@northeastern.edu

[thatha.v@northeastern.edu](mailto:thatha.v@northeastern.edu)

**Percentage of Effort Contributed by Student 1: 50%**

**Percentage of Effort Contributed by Student 2: 50%**

**Signature of Student 1: Dennis Mathew Jose**

**Signature of Student 2: Vamsinath Thatha**

**Submission Date: 02/27/2025**

**Problem 1: Solution**

|  |  |  |
| --- | --- | --- |
| **Sample** | **Probability of New Variant (NV) from the testing laboratory** | **Actual Class** |
| 34 | 0.99 | NV |
| 3 | 0.93 | NV |
| 20 | 0.91 | O |
| 31 | 0.90 | NV |
| 27 | 0.88 | O |
| 23 | 0.87 | NV |
| 10 | 0.81 | NV |
| 21 | 0.80 | NV |
| 22 | 0.79 | O |
| 32 | 0.79 | NV |
| 30 | 0.79 | O |
| 2 | 0.77 | NV |
| 6 | 0.77 | NV |
| 33 | 0.68 | NV |
| 28 | 0.58 | NV |
| 17 | 0.49 | NV |
| 14 | 0.47 | O |
| 16 | 0.46 | O |
| 9 | 0.42 | NV |
| 26 | 0.40 | O |
| 1 | 0.34 | O |
| 4 | 0.33 | NV |
| 18 | 0.32 | O |
| 5 | 0.28 | O |
| 24 | 0.27 | O |
| 25 | 0.26 | O |
| 11 | 0.26 | O |
| 19 | 0.25 | O |
| 29 | 0.24 | NV |
| 8 | 0.18 | O |
| 12 | 0.15 | O |
| 13 | 0.13 | O |
| 7 | 0.09 | O |
| 15 | 0.08 | O |

Here the data contains 34 samples, which show 19 Omicron variants and 15 New variants.

**Problem 1.a)**

The first thing to do is to sort the probabilities in the descending order. Here it is given that the class of interest is the New Variant, which means the positive class C1 is the New Variant and negative class C2 is Omicron.

**When the cut-off is 0:**

Since the least probability is 0.08 when the cut-off is 0.0, every result will be predicted to New Variant but from the data 19 of them have the actual class as Omicron, so these 19 will be classified as False Positive. That is there are 19 values which are classified as New variants even if it is not a new variant.

The confusion matrix is: (Let C1 be the New Variant and C2 be the Omicron variant)

|  |  |  |
| --- | --- | --- |
| For cut-off = 0 |  |  |
|  | Predicted Class  C1 | Predicted Class  C2 |
| Actual Class C1 | 15 (TP) | 0 (FN) |
| Actual Class C2 | 19 (FP) | 0 (TN) |

**For the cut-off 0.2**

When the cut-off is set to 0.2, there are 29 values greater than 0.2. According to the prediction rule, if the probability is below 0.2, the instance is classified as the Omicron variant.

In the sorted table, there are 5 values below 0.2, and the actual class for these records is the Omicron variant (‘O’). Therefore, these can be classified as True Negatives, where the actual class and the predicted class are both Omicron.

Since there are only 5 True Negative values, the remaining actual Omicron variants are misclassified as the New Variant, which are considered False Positives. The number of False Positives is 14.

The confusion matrix is :

|  |  |  |
| --- | --- | --- |
| For cut-off = 0.2 |  |  |
|  | Predicted Class  C1 = NV | Predicted Class  C2 = O |
| Actual Class C1 = O | 15 | 0 |
| Actual Class C2 = NV | 14 | 5 |

**When the cut-off is 0.4:**

Out of the **15 actual New Variant (NV) records, 13** have probability values above **0.4**, meaning they are correctly classified as **New Variant**. The remaining **2** are misclassified as **Omicron**, making them **False Negatives (FN)**—i.e., they are actually New Variant but predicted as Omicron.

Among the **19 actual Omicron (O) variants**, **13** have probability values below **0.4**, so they are correctly classified as **Omicron**. However, the remaining **6**, which have probabilities above **0.4**, are misclassified as **New Variant**, making them **False Positives (FP).**

**Note: Here TN is 13 because the value 0.4 is actually 0.395223183281557 in the Excel file which is adjusted to 0.4. Here we have taken it as 0.39 itself.**

The Confusion Matrix is:

|  |  |  |
| --- | --- | --- |
| For cut-off = 0.4 |  |  |
|  | Predicted Class  C1 = NV | Predicted Class  C2 = O |
| Actual Class C1 = 1 | 13 (TP) | 2 (FN) |
| Actual Class C2 = 0 | 6 (FP) | 13 (TN) |

Similarly, we can find the confusion matrix for the threshold 0.5, 0.6, 0.8 and 1.

**When cut-off is 0.5**:

|  |  |  |
| --- | --- | --- |
| For cut-off = 0.5 |  |  |
|  | Predicted Class  C1 = NV | Predicted Class  C2 = O |
| Actual Class C1 = 1 | 11 | 4 |
| Actual Class C2 = 0 | 4 | 15 |

**When cut-off is 0.6**

|  |  |  |
| --- | --- | --- |
| For cut-off = 0.6 |  |  |
|  | Predicted Class  C1 = NV | Predicted Class  C2 = O |
| Actual Class C1 = 1 | 10 | 5 |
| Actual Class C2 = 0 | 4 | 15 |

**When cut-off is 0.8**

|  |  |  |
| --- | --- | --- |
| For cut-off = 0.8 |  |  |
|  | Predicted Class  C1 = NV | Predicted Class  C2 = O |
| Actual Class C1 = 1 | 6 | 9 |
| Actual Class C2 = 0 | 2 | 17 |

**When cut-off is 1:**

When the cut-off is 1, the scenario is similar to having a cut-off of 0. Since the maximum probability in the dataset is 0.99, which is below 1, every case is classified as Omicron.

As a result, all 15 actual New Variant (NV) records are misclassified as Omicron, making them false negatives. Meanwhile, all 19 actual Omicron (O) variants are correctly classified as Omicron.

The confusion matrix is:

|  |  |  |
| --- | --- | --- |
| For cut-off = 1 |  |  |
|  | Predicted Class  C1 = NV | Predicted Class  C2 = O |
| Actual Class C1 = 1 | 0 | 15 |
| Actual Class C2 = 0 | 0 | 19 |

With the help of this confusion matrix, we can find the sensitivity and specificity for each threshold.

Calculation of sensitivity and specificity for each cut-off

|  |  |  |  |
| --- | --- | --- | --- |
| When cut-off is 0  Sensitivity = 15/(15+0 =1  Specificity = 0/(19+0)=0  When cut-off is 0.6  Sensitivity = 10/15 = 0.667  Specificity = 15/19 = 0.789 | When cut-off is 0.2  Sensitivity = 15/(15+0) = 15  Specificity = 5/19 = 0.263  When cut-off is 0.8  Sensitivity = 6/15 = 0.4  Specificity = 17/19 = 0.894 | When cut-off is 0.4  Sensitivity = 13/15 = .867  Specificity = 13/19 = 0.684  When cut-off is 1  Sensitivity = 0/15 = 0  Specificity = 19/19 = 1 | When cut-off is 0.5  Sensitivity = 11/15 = 0.733  Specificity = 15/19 =0.789 |

|  |  |  |  |
| --- | --- | --- | --- |
| Cut Off | Sensitivity | Specificity | 1-Specificity |
| 0 | 1 | 0 | 1 |
| 0.2 | 1 | 0.263 | 0.737 |
| 0.4 | 0.867 | 0.684 | 0.316 |
| 0.5 | 0.733 | 0.789 ~ 0.79 | 0.21 |
| 0.6 | 0.666 | 0.789 ~ 0.79 | 0.21 |
| 0.8 | 0.4 | 0.894 | 0.106 |
| 1 | 0 | 1 | 0 |

We can use the data in this table to plot the ROC curve.

Python code to generate ROC Curve:

# Data from the table

cutoffs = [0, 0.2, 0.4, 0.5, 0.6, 0.8, 1]

sensitivity = [1, 1, 0.867, 0.733, 0.666, 0.4, 0]

specificity = [0, 0.263, 0.684, 0.789, 0.789, 0.894, 1]

# Calculate 1 - specificity for the False Positive Rate (FPR)

fpr = 1 - np.array(specificity)

tpr = np.array(sensitivity) # True Positive Rate is the same as Sensitivity

# Plot the ROC curve

plt.figure(figsize=(8, 6))

plt.plot(fpr, tpr, marker='o', linestyle='-', color='b', label='ROC curve')

# Plot diagonal line (random classifier)

plt.plot([0, 1], [0, 1], linestyle='--', color='gray', label='Random classifier')

# Set labels and title

plt.xlabel('False Positive Rate (1-specificity)')

plt.ylabel('True Positive Rate (Sensitivity)')

plt.title('Receiver Operating Characteristic (ROC) Curve')

plt.legend(loc='lower right')

# Show the plot

plt.grid(True)

plt.show()

A graph with a line

Description automatically generated

The optimal cut-off, based on the graph, is the maximum sum of sensitivity and specificity. Here, the optimal cut-off is 0.4 (Sensitivity = 0.864 and specificity = 0.684).

**Problem 1.b**

For the default threshold of 0.5,

The Confusion matrix is:

|  |  |  |
| --- | --- | --- |
| For cut-off = 0.5 |  |  |
|  | Predicted Class  C1 = NV | Predicted Class  C2 = O |
| Actual Class C1 = 1 | 11 (TP) | 4 (FN) |
| Actual Class C2 = 0 | 4 (FP) | 15 (TN) |

Substituting values in MCC

Here the optimal cut-off is 0.4. From the confusion matrix of 0.4

|  |  |  |
| --- | --- | --- |
| For cut-off = 0.4 |  |  |
|  | Predicted Class  C1 = NV | Predicted Class  C2 = O |
| Actual Class C1 = 1 | 13 (TP) | 2 (FN) |
| Actual Class C2 = 0 | 6 (FP) | 13 (TN) |

MCC for optimal cut-off is:

**Note**: Based on the findings the optimal cut-off (0.4) is recommended because it maximizes the MCC, indicating a better balance between sensitivity and specificity. The slight improvement of the cut-off from 0.5 to 0.4 demonstrates that adjusting the cut-off can enhance the model. The higher MCC value reflects a **better balance between sensitivity and specificity**, reducing the number of false positives and false negatives compared to the default cutoff.

performance.

**Problem 2: Solution**

**For the training Set**

Given that Training data contains 100000 eye-scanned radiology films, which shows 40000 Diabetic retinopathy conditions and 60000 normal conditions.

The model classified 38950 Diabetic retinopathy conditions and 58500 other normal

conditions correctly.

The confusion matrix will be:

Since the model classified Diabetic Retinopathy as 38950 out of 40000 Diabetic Retinopathy, The true positive, which is correctly predicted diabetic retinopathy cases, will be 38950.

*i.e.,* TP = 38950

Since TP = 38950, FN = 40000 – 38950 = 1050

The FN is False Negative, which is the case where Diabetic retinopathy cases are incorrectly classified as normal conditions.

Since the model classified the Normal condition as 58500 out of 60000 normal conditions, the true negative, which is correctly predicted normal cases, will be 58500.

*i.e.,* TN = 58500

Since TN = 58500, FP = 60000 – 58500 = 1500

The FP is called False Positive, where the case Normal condition is incorrectly classified as Diabetic Retinopathy

|  |  |  |
| --- | --- | --- |
|  | Predicted Diabetic Retinopathy | Predicted Normal Condition |
| Actual Diabetic Retinopathy | 38950 (TP) | 1050 (FN) |
| Actual Normal Condition | 1500 (FP) | 58500 (TN) |

Based on the Confusion matrix we can calculate:

**For the Validation Set**

Given that the validation set consists of 10000 radiology films (3750 Diabetic retinopathy conditions and 6250 normal conditions).

The model classified 2500 Diabetic retinopathy conditions and 4975 other normal conditions correctly.

The confusion matrix will be

The true positives will be the number of records which the model classified correctly as Diabetic Retinopathy condition, which is 2500 films.

i.e., TP = 2500

Since TP is 2500, the FN (False negatives) = 3750 – 2500 = 1250. The False negatives imply that the case where Diabetic retinopathy cases are incorrectly classified as normal conditions.

Given that the model classified 4975 normal conditions correctly, it implies that the true Negatives are 4975.

i.e., TN = 4975

Since TN = 4975, FP (False Positives) will be 6250 – 4975 = 1275. The false Positives are the cases where the Normal condition is incorrectly classified as Diabetic Retinopathy.

|  |  |  |
| --- | --- | --- |
|  | Predicted Diabetic Retinopathy | Predicted Normal Condition |
| Actual Diabetic Retinopathy | 2500 (TP) | 1250 (FN) |
| Actual Normal Condition | 1275 (FP) | 4975 (TN) |

**Model Performance**

High performance on the training data. The model performs well on the training set, with a very low error rate of 0.0255 or 2.5%. The sensitivity and specificity are both high, at 97.37% and 97.5%, respectively, indicating that the model correctly classifies Diabetic Retinopathy and normal conditions based on the films.

However, there is a significant decline in the performance of the validation set. The error rate rises to 25.25%, indicating an overfitting of the data. Sensitivity decreases to 66.67%, meaning the model is missing numerous cases of diabetic retinopathy, which poses risks in medical applications. The specificity falls to 79.6%, which implies that more cases are misclassified as diabetic retinopathy.

The drop in sensitivity is particularly concerning because missing actual diabetic retinopathy cases can delay diagnosis and treatment leading to blindness. Overfitting of data points is very evident, so the model needs to be improved to address these issues.

**Problem 3:**

**3.a)**

Total Samples-7000, Positive Cases (Type-A Disorder): 2800,

Validation Set Size - 30% of 7000 = 2100 samples

Sensitivity (Recall) - 60%

Specificity - 80%

Positive Cases in Validation Set: 2800×0.3 = 840

True Positives (TP): Sensitivity × Positive Cases = 0.6×840 = 504

False Negatives (FN): Remaining positive cases = 840−504 = 336

Negative Cases in Validation Set: (7000−2800) ×0.3 = 1260

True Negatives (TN): Specificity × Negative Cases = 0.8×1260 = 1008

False Positives (FP): Remaining negative cases = 1260−1008 = 252

Confusion Matrix will be :

|  |  |  |
| --- | --- | --- |
|  | Predicted Class  C1 | Predicted Class  C2 |
| Actual Class C1 | 504 (TP) | 336 (FN) |
| Actual Class C2 | 252 (FP) | 1008 (TN) |

Adjusted misclassification rate = FP+FN/Total samples = 588/2100 = 0.28 (~28%)

**Precision** = TP/(TP+FP) = 504/756 = 0.67 (~67%)

**Recall (Sensitivity)** = 60%

**Analysis on model performance:**

**Moderate Sensitivity (60%):** The model detects only 60% of actual Type-A disorder cases, meaning it misses 40% of true cases. This could be a significant issue for a medical diagnosis model, as failing to identify a large proportion of true cases may lead to untreated conditions and poor patient outcomes.

**High Specificity (80%):** The model correctly identifies 80% of negative cases, but 20% of healthy individuals are misclassified as having the disorder. While the specificity is relatively high, the misclassification of healthy individuals could result in unnecessary stress, additional testing, and increased healthcare costs.

**Precision (66.7%):** Among the cases predicted as positive, only 66.7% are truly positive. This level of precision is not ideal for a medical screening tool, as it means a significant portion of individuals flagged as positive do not actually have the disorder, potentially leading to overdiagnosis and wasted resources.

**Misclassification Rate (28%):** The model has an overall misclassification rate of 28%, meaning 28% of its total predictions are incorrect. This is relatively high for medical applications, where accuracy and reliability are critical to ensure proper diagnosis and treatment.

**3.b** Since only 40% of the dataset consists of positive cases (2800), the class distribution is imbalanced. Therefore, Oversampling the minority class, undersampling the majority class and using different evaluation metrics like AUC or F1-score can help.

**Problem 4:**

Net Profit = Service cost – (Data engineer + BI Engineer + Solutions architect + Sales engineer + Project manager) = Service cost – 335

1. Since the dataset is large, it is not included in this document. The link below provides access to an Excel sheet, where the gain chart has been created step by step  
   [Gain Chart and Calculations:](https://northeastern-my.sharepoint.com/:x:/g/personal/jose_de_northeastern_edu/Efcl8PO5QA5NtkzrHyZLh0kBa4o479QVrqPMvVedSMsonw?e=xOHu6u)  
     
   A graph with blue and orange lines

   Description automatically generated
2. The top 10 clients to contact are [Amazon.com](http://amazon.com/), Charter Communications, AIG, MetLife, Citigroup, Bank of America, PepsiCo, Morgan Stanley, Ford Motor, and Home Depot.

Total Cost for 1 Call = $60 + $75 + $100 + $50 + $50 = $335

Cost of 10 Calls = 10 x $335 = $3,350

Total Profit: $1500 + $618 + $490 + $948 + $702 + $1000 + $827 + $644 + $953 + $525 = $8,207

The sum of the net profits of the top 10 clients: $8,207 - $3,350 = $4,857

Average Net profit per client = Total profit/no of clients = 28003/72 = $388.9

Baseline profit for 10 clients = $388.9\*10 = $3,889

Gain compared to baseline = $4,857-$3,889 = $968

1. USA-based companies using "GCP Big Query" as their data warehouse and "Looker" as their BI tool tend to have a higher probability of winning opportunities. Allocate more resources and focus more on to regions and tools that are common among top clients (e.g., USA, GCP Big Query, Looker). Cloud storage size and service cost do not show a clear correlation with opportunity probability. So, other business factors may probably influence this.