# **LM 7 – Assignment**

1. **When AUC can be used as a measure of quality of models? How does it differ from other measures such as accuracy, precision or recall?**

**Ans:** AUC (Area Under the Curve) can be used to assess model quality under specific conditions and offers a different perspective than other measures such as Accuracy, Precision, and Recall. Let me give you an overview:  
When to Use AUC:

* AUC is a valuable metric for assessing models on imbalanced datasets, when one class is much more prevalent than the other. This is frequently observed in situations such as fraud detection and disease screening.
* The area under the curve (AUC) is a measurement that determines the quality of the model's predictions regardless of any specific categorization threshold. It assesses the model's capacity to differentiate between positive and negative categories at all potential levels.
* AUC is useful when the model produces probabilistic predictions (i.e., the likelihood of belonging to a specific class), because it measures how effectively the model can rank positive samples higher than negative ones.

How AUC differs from Accuracy, Precision, and Recall Accuracy:

* Accuracy: The metric measures the percentage of accurate outcomes (including both correctly identified positives and negatives) to the overall number of cases evaluated. The accuracy of a model with imbalanced datasets might be deceptive because it might primarily represent the distribution of classes rather than the model's capability to distinguish between different classes.
* Precision: The percentage of positive identifications that were right. The precision of the positive predictions is especially valuable when the consequences of a false positive are significant.
* Recall (Sensitivity): The percentage of true positives that were accurately identified. This statistic is essential when the potential consequences of failing to detect a positive result (false negative) are significant.

The AUC (Area Under the Curve) is a metric that quantifies the ability of a model to differentiate between classes, regardless of the distribution of the classes or the exact decision threshold used. Unlike accuracy and recall, which are measurements that only consider a single threshold, AUC (Area Under the Curve) evaluates the performance of the model across all possible thresholds.

1. **When designing a system for detecting life-threatening events for ICU patients, what is more important: precision or recall? Why?**

**Ans:** When building a system for detecting possibly life-threatening events in ICU patients, recall takes importance over precision. Here’s why this prioritization is essential:

Understanding Precision and Recall:

* Precision: Refers to the percentage of accurate positive forecasts—the proportion of alerts that are true positives.
* Recall: This measures the percentage of all positive life-threatening events that the model correctly detects. A system with high recall will accurately identify most or all the life-threatening incidents, even if it also results in some false positives.

High Recall System:

* Broad Detection: A system with high recall will identify most or all life-threatening occurrences. This extensive coverage is crucial in the ICU, where failing to detect a critical condition can have dire consequences.
* Acceptance of False Positives: While such a system may also identify some false positives, this is preferable to missing actual life-threatening events.

High Precision System:

* Limited Detection: Although a system with high precision will correctly identify mostly true positives, its stringent criteria may lead it to miss certain life-threatening situations.
* Risk of Missed Threats: Missing a critical event because the system was too conservative in its alerts could delay necessary interventions, endangering patient health.

Consequences of System Performance:

* Managing False Positives: In instances of false positives, the clinical team can conduct further investigations to verify the alerts, thus confirming or dismissing the potential threat.
* Dangers of Missed Alerts: Conversely, if a life-threatening event is missed by the system, the patient may not receive the needed care promptly. This oversight could result in severe illness or even death, as timely medical intervention is critical.

Key area of focus:

* Safety Over Inconvenience: In the ICU context, the repercussions of missing a potentially fatal event are far more severe than those of managing false positives.
* Clinical Verification: False alarms can be further assessed by the clinical team, ensuring that they do not lead to unnecessary procedures but also safeguard against oversight.

In conclusion, when building systems for monitoring possibly life-threatening events in intensive care unit settings, recall should take importance over precision. This method captures all life-threatening occurrences, lowering the likelihood of undiagnosed conditions causing fatalities. False positives are controllable compared to missing life-threatening occurrences.

1. **Using testClaims data prepared last week**
2. **Include drug information into data. There are many ways this can be accomplished. Propose a reasonable solution and execute it.**
   1. **Include procedure information into data. There are many ways this can be accomplished. Propose a reasonable solution and execute it.**

**Ans:** This is the code I executed in SQL:

use Test\_claims

-- Create Elix comorbidity indicators for each patient

SELECT c.patient\_id

,max(case when [diagnosis] = 'ELIX1' then 1 else 0 end) as ELIX1

,max(case when [diagnosis] = 'ELIX10' then 1 else 0 end) as ELIX10

,max(case when [diagnosis] = 'ELIX11' then 1 else 0 end) as ELIX11

,max(case when [diagnosis] = 'ELIX12' then 1 else 0 end) as ELIX12

,max(case when [diagnosis] = 'ELIX13' then 1 else 0 end) as ELIX13

,max(case when [diagnosis] = 'ELIX14' then 1 else 0 end) as ELIX14

,max(case when [diagnosis] = 'ELIX15' then 1 else 0 end) as ELIX15

,max(case when [diagnosis] = 'ELIX16' then 1 else 0 end) as ELIX16

,max(case when [diagnosis] = 'ELIX17' then 1 else 0 end) as ELIX17

,max(case when [diagnosis] = 'ELIX18' then 1 else 0 end) as ELIX18

,max(case when [diagnosis] = 'ELIX19' then 1 else 0 end) as ELIX19

,max(case when [diagnosis] = 'ELIX2' then 1 else 0 end) as ELIX2

,max(case when [diagnosis] = 'ELIX20' then 1 else 0 end) as ELIX20

,max(case when [diagnosis] = 'ELIX21' then 1 else 0 end) as ELIX21

,max(case when [diagnosis] = 'ELIX22' then 1 else 0 end) as ELIX22

,max(case when [diagnosis] = 'ELIX23' then 1 else 0 end) as ELIX23

,max(case when [diagnosis] = 'ELIX24' then 1 else 0 end) as ELIX24

,max(case when [diagnosis] = 'ELIX25' then 1 else 0 end) as ELIX25

,max(case when [diagnosis] = 'ELIX26' then 1 else 0 end) as ELIX26

,max(case when [diagnosis] = 'ELIX27' then 1 else 0 end) as ELIX27

,max(case when [diagnosis] = 'ELIX28' then 1 else 0 end) as ELIX28

,max(case when [diagnosis] = 'ELIX29' then 1 else 0 end) as ELIX29

,max(case when [diagnosis] = 'ELIX3' then 1 else 0 end) as ELIX3

,max(case when [diagnosis] = 'ELIX4' then 1 else 0 end) as ELIX4

,max(case when [diagnosis] = 'ELIX5' then 1 else 0 end) as ELIX5

,max(case when [diagnosis] = 'ELIX6' then 1 else 0 end) as ELIX6

,max(case when [diagnosis] = 'ELIX7' then 1 else 0 end) as ELIX7

,max(case when [diagnosis] = 'ELIX8' then 1 else 0 end) as ELIX8

,max(case when [diagnosis] = 'ELIX9' then 1 else 0 end) as ELIX9

INTO #ELIX FROM dbo.diagnoses d

JOIN dbo.claims c ON d.claim\_id = c.claim\_id

GROUP BY c.patient\_id -----(145929 rows affected)

---Step 2:-- Aggregate procedure information by patient and format hcpcs\_grp

SELECT c.patient\_id,

REPLACE(p.hcpcs\_grp, '-', '') AS hcpcs\_grp,

COUNT(p.hcpcs) AS procedure\_count

INTO #PROCEDURES FROM dbo.[procedures] p

JOIN dbo.claims c ON p.claim\_id = c.claim\_id

GROUP BY c.patient\_id, REPLACE(p.hcpcs\_grp, '-', '') ---(810997 rows affected)

select \* from #PROCEDURES

---Step 3: -- Define high utilizers based on claim counts in year 'Y2'

SELECT patient\_id,

COUNT(\*) AS CLAIMS,

CASE WHEN COUNT(\*) >= 100 THEN 1 ELSE 0 END AS highUtilizer

INTO #HIGH\_UTILIZERS FROM dbo.claims WHERE year = 'Y2'

GROUP BY patient\_id ---(114663 rows effected)

select \* from #HIGH\_UTILIZERS

---Step 4: -- Combine all data into a final dataset

SELECT e.patient\_id,

e.ELIX1, e.ELIX2, e.ELIX3, e.ELIX4, e.ELIX5, e.ELIX6, e.ELIX7, e.ELIX8, e.ELIX9, e.ELIX10,

e.ELIX11, e.ELIX12, e.ELIX13, e.ELIX14, e.ELIX15, e.ELIX16, e.ELIX17, e.ELIX18, e.ELIX19, e.ELIX20,

e.ELIX21, e.ELIX22, e.ELIX23, e.ELIX24, e.ELIX25, e.ELIX26, e.ELIX27, e.ELIX28, e.ELIX29,

p.hcpcs\_grp, p.procedure\_count,

h.highUtilizer

INTO dbo.procedures\_classification\_final

FROM #ELIX e

LEFT JOIN #PROCEDURES p ON e.patient\_id = p.patient\_id

LEFT JOIN #HIGH\_UTILIZERS h ON e.patient\_id = h.patient\_id

select \* from dbo.procedures\_classification\_final

-- Select the top 100,000 rows from the dbo.procedures\_classification\_final

SELECT TOP 100000

patient\_id,

ELIX1, ELIX2, ELIX3, ELIX4, ELIX5, ELIX6, ELIX7, ELIX8, ELIX9, ELIX10,

ELIX11, ELIX12, ELIX13, ELIX14, ELIX15, ELIX16, ELIX17, ELIX18, ELIX19, ELIX20,

ELIX21, ELIX22, ELIX23, ELIX24, ELIX25, ELIX26, ELIX27, ELIX28, ELIX29,

hcpcs\_grp,procedure\_count,highUtilizer

INTO dbo.final\_classification\_data\_subset

FROM [dbo].[procedures\_classification\_final]

ORDER BY patient\_id

select \* from dbo.final\_classification\_data\_subsetA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**I am selecting only the top 1,00,000 rows for weka.**

A screenshot of a computer

Description automatically generated

* 1. **Import newly created data into Weka and build classification models.**

**Ans:** I am taking only 1,00,000 rows for weka.Now, I loaded thedbo.final\_classification\_data\_subset to weka as final\_classification\_data\_subset.

**A screenshot of a computer

Description automatically generated**

**Logistic Regression Model:**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Random Forest:**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**J48:**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

* 1. **Does inclusion of the information improve predictions?**

**Ans:** Indeed, the use of specific data points such as ELIX codes and HCPCS groups greatly enhances the accuracy of forecasts. Upon analyzing the ROC Area outcomes of several models evaluated in Weka, it is evident that models, particularly the Random Forest, exhibit improved predicted precision when provided with this supplementary data. The Random Forest model achieved a ROC Area of 0.992 for class 1, indicating a good level of accuracy in distinguishing between high utilizers and non-high utilizers. The Random Forest model efficiently utilizes the rich dataset, including comprehensive comorbidity and procedural information, to capture complex interactions and non-linear connections, as evidenced by the large ROC Area. Thus, the inclusion of extensive data can greatly enhance the efficiency of predictive models, especially those that can handle intricate data structures, thereby assisting in improved healthcare management and decision-making.

* 1. **The constructed models are built from Year 1 data to predict outcome in Year 2. Now consider the possibility of applying the model to data from Year 2 to predict Year 3. Prepare data to do so, and execute the test. What is the performance of the model as compared to original testing on prediction of year 2?**

**Ans:** USE Test\_claims

---step 1: create list of patients

select patient\_id into #patients\_Y2\_Y3

from dbo.claims where year = 'Y2' or year = 'Y3'

group by patient\_id having min(year) != max(year) --- (94362 rows affected)

----Step 2: creating binary ELIX

SELECT c.patient\_id,

MAX(CASE WHEN d.diagnosis = 'ELIX1' THEN 1 ELSE 0 END) AS ELIX1,

MAX(CASE WHEN d.diagnosis = 'ELIX2' THEN 1 ELSE 0 END) AS ELIX2,

MAX(CASE WHEN d.diagnosis = 'ELIX3' THEN 1 ELSE 0 END) AS ELIX3,

MAX(CASE WHEN d.diagnosis = 'ELIX4' THEN 1 ELSE 0 END) AS ELIX4,

MAX(CASE WHEN d.diagnosis = 'ELIX5' THEN 1 ELSE 0 END) AS ELIX5,

MAX(CASE WHEN d.diagnosis = 'ELIX6' THEN 1 ELSE 0 END) AS ELIX6,

MAX(CASE WHEN d.diagnosis = 'ELIX7' THEN 1 ELSE 0 END) AS ELIX7,

MAX(CASE WHEN d.diagnosis = 'ELIX8' THEN 1 ELSE 0 END) AS ELIX8,

MAX(CASE WHEN d.diagnosis = 'ELIX9' THEN 1 ELSE 0 END) AS ELIX9,

MAX(CASE WHEN d.diagnosis = 'ELIX10' THEN 1 ELSE 0 END) AS ELIX10,

MAX(CASE WHEN d.diagnosis = 'ELIX11' THEN 1 ELSE 0 END) AS ELIX11,

MAX(CASE WHEN d.diagnosis = 'ELIX12' THEN 1 ELSE 0 END) AS ELIX12,

MAX(CASE WHEN d.diagnosis = 'ELIX13' THEN 1 ELSE 0 END) AS ELIX13,

MAX(CASE WHEN d.diagnosis = 'ELIX14' THEN 1 ELSE 0 END) AS ELIX14,

MAX(CASE WHEN d.diagnosis = 'ELIX15' THEN 1 ELSE 0 END) AS ELIX15,

MAX(CASE WHEN d.diagnosis = 'ELIX16' THEN 1 ELSE 0 END) AS ELIX16,

MAX(CASE WHEN d.diagnosis = 'ELIX17' THEN 1 ELSE 0 END) AS ELIX17,

MAX(CASE WHEN d.diagnosis = 'ELIX18' THEN 1 ELSE 0 END) AS ELIX18,

MAX(CASE WHEN d.diagnosis = 'ELIX19' THEN 1 ELSE 0 END) AS ELIX19,

MAX(CASE WHEN d.diagnosis = 'ELIX20' THEN 1 ELSE 0 END) AS ELIX20,

MAX(CASE WHEN d.diagnosis = 'ELIX21' THEN 1 ELSE 0 END) AS ELIX21,

MAX(CASE WHEN d.diagnosis = 'ELIX22' THEN 1 ELSE 0 END) AS ELIX22,

MAX(CASE WHEN d.diagnosis = 'ELIX23' THEN 1 ELSE 0 END) AS ELIX23,

MAX(CASE WHEN d.diagnosis = 'ELIX24' THEN 1 ELSE 0 END) AS ELIX24,

MAX(CASE WHEN d.diagnosis = 'ELIX25' THEN 1 ELSE 0 END) AS ELIX25,

MAX(CASE WHEN d.diagnosis = 'ELIX26' THEN 1 ELSE 0 END) AS ELIX26,

MAX(CASE WHEN d.diagnosis = 'ELIX27' THEN 1 ELSE 0 END) AS ELIX27,

MAX(CASE WHEN d.diagnosis = 'ELIX28' THEN 1 ELSE 0 END) AS ELIX28,

MAX(CASE WHEN d.diagnosis = 'ELIX29' THEN 1 ELSE 0 END) AS ELIX29

INTO #ELIX\_Y2 FROM [dbo].[claims] c JOIN [dbo].[diagnoses] d ON c.claim\_id = d.claim\_id

WHERE c.year = 'Y2' GROUP BY c.patient\_id;

select \* from #ELIX\_Y2

--step 3: create the dependent variable

select patient\_id, count(\*) as num\_claims,

case when count(\*) >= 100 then 1 else 0 end as Highutilizer

into #Num\_ClaimY from dbo.claims where year = 'Y3'

group by patient\_id----(110942 rows affected)

select \* from #Num\_ClaimY

--step 4: Finally join the tables

select #ELIX\_Y2.\* into #t1 from #patients\_Y2\_Y3,

#ELIX\_Y2 where #patients\_Y2\_Y3.patient\_id = #ELIX\_Y2.patient\_id --(94362 rows affected)

select #t1.\*, num\_claims, Highutilizer

into dbo.HighutilizationYr3

from #t1, #Num\_ClaimY

where #t1.patient\_id = #Num\_ClaimY.patient\_id

select \* from dbo.HighutilizationYr3

A screenshot of a computer

Description automatically generated

Logistic regression Model:

Cross validation – 10 folds

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

References:

* Blackboard material.
* <https://www.shiksha.com/online-courses/articles/roc-auc-vs-accuracy/#:~:text=Generally%2C%20AUC%20is%20preferred%20over,use%20other%20metrics%20as%20well>.
* <https://www.nature.com/articles/s41591-020-0789-4>