Predicting Mortality in Sepsis-Associated Acute Respiratory Distress Syndrome (ARDS)

Rithvik V.Sourab

Lekhana Chandra Palamuri

Wenjia Duan

Introduction

ARDS is a critical illness associated with sepsis, and its early prediction is crucial for improving clinical outcomes. Despite advancements in medical care, ARDS-related mortality remains high(30%-50%), requiring the development of robust prediction models for better resource allocation and clinical decision-making.

This research aims to build and validate a machine learning model to predict mortality in patients with Sepsis-Associated ARDS, leveraging the MIMIC-III database.

Data collection

1. Patients: get demographic information
2. Admissions: Extract admission details such as admission time, admission type
3. Icustays: filter icustay<=48
4. diagnoses_icd /d_icd_diagnoses: find patients diagnosed with ARDS
5. Chartevents/d_items: Extract clinical data and other bedside monitoring information.
6. Labevents/d_labitems: Obtain laboratory test results

Use HADM_ID join all table together

SQL Code breakdown:

Step 1,2:

```
SELECT
A.SUBJECT_ID, A.DOB,
B.HADM_ID, B.ADMITTIME ADMITTIME, B.ADMISSION_TYPE, B.INSURANCE, B.DIAGNOSIS
FROM
`physionet-data.mimiciii_clinical.patients` AS A
LEFT JOIN
`physionet-data.mimiciii_clinical.admissions` AS B
ON
A.SUBJECT_ID = B.SUBJECT_ID where A.SUBJECT_ID in (SELECT subject_id
```

Step 3:

```
FROM
`physionet-data.mimiciii_clinical.icustays`
WHERE date_diff(outtime,intime,day)<2 or date_diff(outtime,intime,day)=2
```

Step 4:

```
SELECT
C2.HADM_ID, C2.SUBJECT_ID
FROM
`physionet-data.mimiciii_clinical.diagnoses_icd` AS C2
LEFT JOIN
`physionet-data.mimiciii_clinical.d_icd_diagnoses` AS D2
ON
C2.ICD9_CODE = D2.ICD9_CODE
WHERE
C2.ICD9_CODE IN ('99591', '99592', '0389')
),
```

Step 5:

```
SELECT
E.HADM_ID, E.ICUSTAY_ID, AVG(E.valuenum) AS value, di.label
FROM
`physionet-data.mimiciii_clinical.chartevents` AS E
JOIN
`physionet-data.mimiciii_clinical.d_items` di
ON
E.itemid = di.itemid
WHERE
E.ITEMID IN (224422, 618, 220210, 224689, 614, 651, 224690, 615, 211, 220045, 442, 227243, 224167, 220179, 225309, 646, 834, 220277,223761,678,226707,763, 224639,
226512,198, 199, 200, 220210, 51265, 220180, 221185, 221179, 221180, 221181, 221182, 1040,1206,189,190,191,720, 223848, 467, 445, 224832,456, 52, 220052, 220181)
GROUP BY E.HADM_ID, E.ICUSTAY_ID, di.label
),
```

Step 6:

```
SELECT
G.HADM_ID, G.ITEMID, G.VALUE, G.VALUENUM, H.LABEL, H.CATEGORY
FROM
`physionet-data.mimiciii_clinical.labevents` AS G
LEFT JOIN
`physionet-data.mimiciii_clinical.d_labitems` AS H
ON
G.ITEMID = H.ITEMID
WHERE
H.LABEL LIKE 'Glucose%'
OR H.LABEL LIKE 'Lactate%'
OR H.LABEL LIKE 'Potassium%'
OR H.LABEL LIKE 'Albumin%'
OR H.LABEL LIKE 'Bilirubin%'
OR H.LABEL LIKE 'Creatinine%'
OR H.LABEL LIKE 'Chloride%'
OR H.LABEL LIKE 'pCO2%'
OR H.LABEL LIKE 'Urea Nitrogen%'
OR H.LABEL LIKE 'Urine%'
OR H.LABEL LIKE 'Platelet Count'
OR H.LABEL LIKE 'pO2'),
```

Calculating SOFA Score:

Convert long-format to wide-format
calculate clinical index depend on exist data

Example Calculation of SOFA Score

Here's a simplified overview of how the individual components of the SOFA score are calculated:

Organ System	Score	Criteria
Respiratory	0	PaO2/FiO2 > 400
	1	PaO2/FiO2 300-400
	2	PaO2/FiO2 200-300
	3	PaO2/FiO2 100-200
	4	PaO2/FiO2 < 100 or on mechanical ventilation
Coagulation	0	Platelet count > 150,000
	1	Platelet count 100,000-150,000
	2	Platelet count 50,000-100,000
	3	Platelet count 20,000-50,000
	4	Platelet count < 20,000
Liver	0	Bilirubin < 1.2 mg/dL
	1	Bilirubin 1.2-1.9 mg/dL
	2	Bilirubin 2 [illegible].9 mg/dL
	3	Bilirubin 6.0-12.0 mg/dL

	4	Any dose of vasopressors
Renal	0	Creatinine < 1.2 mg/dL or urine output > 0.5 mL/kg/h
	1	Creatinine 1.2-1.9 mg/dL or urine output < 0.5 mL/kg/h
	2	Creatinine 2.0-3.4 mg/dL or urine output < 0.3 mL/kg/h
	3	Creatinine 3.5 mg/dL or urine output < 0.3 mL/kg/h
	4	Renal replacement therapy
Neurological	0	GCS 15
	1	GCS 13-14
	2	GCS 10-12
	3	GCS 6-9
	4	GCS < 6

Data Extraction - Count: 2799

Data cleaning

Filling the missing values:

- BMI mean and median imputation within range (27-30) replaced with NaN Values.
- Removed missing values from pt and ptt average.

```
BMI Mean for Imputation: 28.3446942835
BMI Median for Imputation: 28.22894606
         bmi  bmi_mean_imputed  bmi_median_imputed
0        NaN         28.344694           28.228946
1  17.395254         17.395254           17.395254
2        NaN         28.344694           28.228946
3  16.975309         16.975309           16.975309
4        NaN         28.344694           28.228946
```

- Removed rows where age is >=300

```
Number of rows removed: 63
Cleaned dataset saved to: pt and ptt_dataset.csv
```

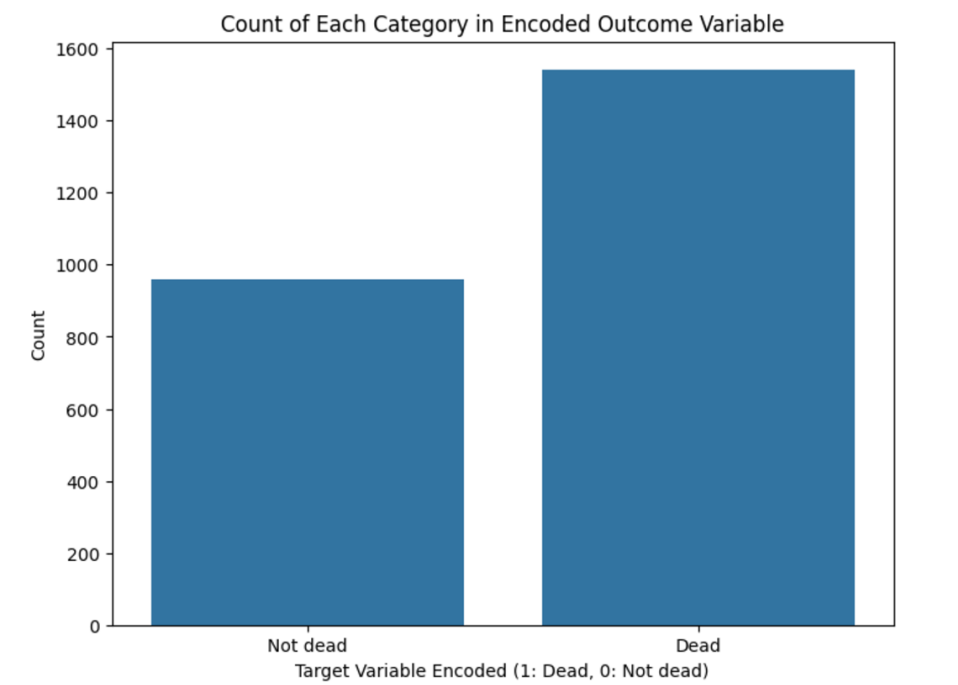
- convert any categorical outcomes appropriately

```
Number of rows removed: 234
```

- After Data Cleaning - Count: 2502

Outcome Variable Statistics

- Calculated target variable using EXPIRE_FLAG, Whether a patient died during their hospital stay(1-Dead, 0-Not dead)



Target_variable	
Dead	1542
Not dead	959

Data Statistics: Mean, Median, Missing Value count for independent and dependent variables.

Basic Data Statistics for Independent and Outcome Variables:

	Variable	Mean	Median	Missing Values
0	age	65.957617	68.000000	0
1	abpd	0.329825	0.000000	0
2	bilirubin_avg	0.134807	0.019048	0
3	lactate_avg	14.507750	3.873810	0
4	glucose_avg	17.114161	15.368421	0
5	potassium_avg	0.627020	0.548438	0
6	creatinine_avg	0.735462	0.313462	0
7	urine_avg	0.107800	0.000000	0
8	albumin_avg	0.088416	0.051724	0
9	bun_avg	4.845672	3.340796	0
10	pCO2_avg	2.209041	1.385787	0
11	valvular_disease	0.020792	0.000000	0
12	metastatic_cancer	0.020392	0.000000	0
13	pt_avg	29.427373	25.414286	0
14	ptt_avg	40.736533	35.150000	0
15	bmi_mean_imputed	111.404706	28.344694	0
16	bmi_median_imputed	111.325057	28.228946	0
17	Target_variable_encoded	0.616553	1.000000	0

Data Splitting & Model Development

Trained seven machine learning models:

Logistic Regression, Random Forest, Naive Bayes, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Decision Tree, and XG Boost.

Randomly split data into training (70%) and testing (30%) cohorts. Ensured balanced data distribution between training and test sets.

```
Data Splitting Summary:
Training Set Size: 1750
Testing Set Size: 751
Training Class Balance:
Target_variable_encoded
1    0.616571
0    0.383429
Name: proportion, dtype: float64
Testing Class Balance:
Target_variable_encoded
1    0.616511
0    0.383489
```

Model Evaluation

- Performance metrics included AUC (Area Under the Curve), accuracy, sensitivity, specificity, positive predictive value (PPV), and negative predictive value (NPV).

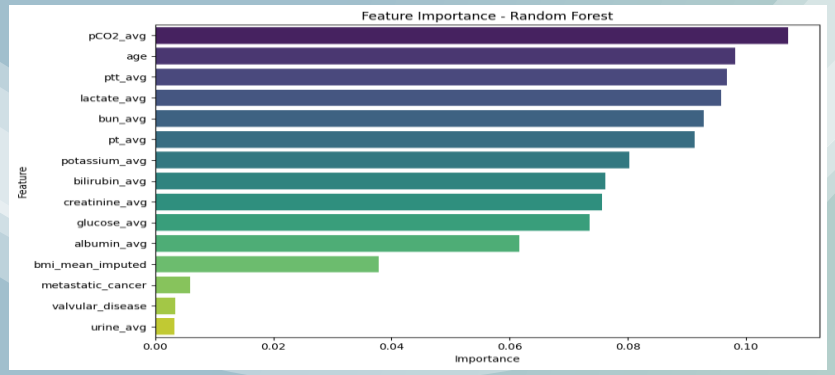
Train and Test Evaluation Metrics:

	Model	Train Accuracy	Test Accuracy	Train AUC	Test AUC
0	Logistic Regression	0.720000	0.700399	0.783500	0.748830
1	Random Forest	1.000000	0.703063	1.000000	0.741559
2	XGBoost	1.000000	0.697736	1.000000	0.741526
3	Naive Bayes	0.683429	0.661784	0.740723	0.695719
4	K-Nearest Neighbors	0.770286	0.657790	0.840191	0.692397
5	Support Vector Machine	0.617143	0.616511	0.734027	0.698907
6	Decision Tree	1.000000	0.653795	1.000000	0.639166

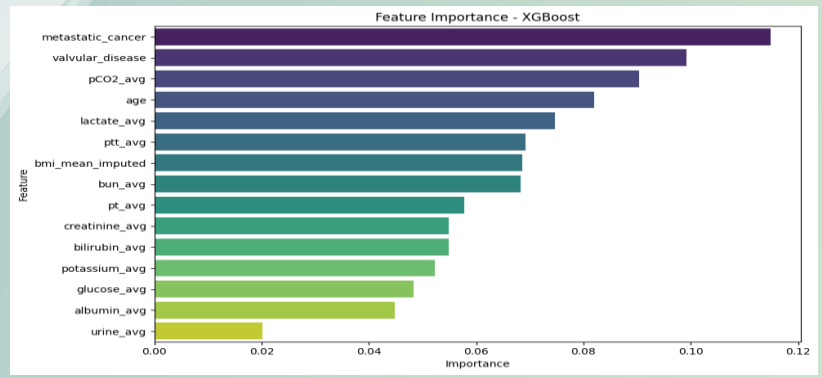
	Train Sensitivity	Test Sensitivity	Train Specificity	Test Specificity
0	0.821131	0.805616	0.557377	0.531250
1	1.000000	0.796976	1.000000	0.552083
2	1.000000	0.796976	1.000000	0.538194
3	0.917516	0.896328	0.307004	0.284722
4	0.846154	0.753780	0.648286	0.503472
5	1.000000	1.000000	0.001490	0.000000
6	1.000000	0.701944	1.000000	0.576389

	Train PPV	Test PPV	Train NPV	Test NPV
0	0.748943	0.734252	0.659612	0.629630
1	1.000000	0.740964	1.000000	0.628458
2	1.000000	0.735060	1.000000	0.622490
3	0.680412	0.668277	0.698305	0.630769
4	0.794604	0.709350	0.723794	0.559846
5	0.616924	0.616511	1.000000	NaN
6	1.000000	0.727069	1.000000	0.546053

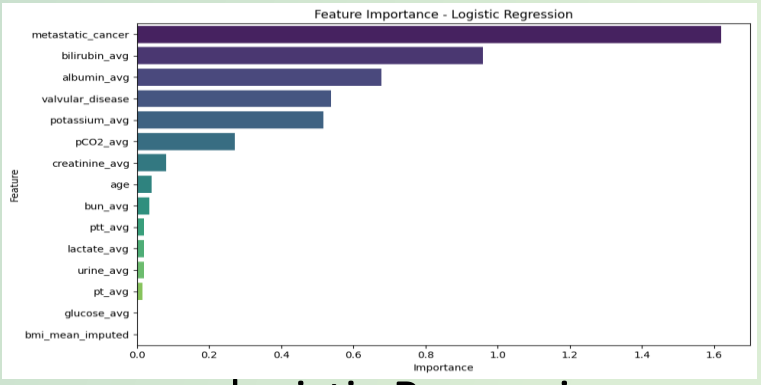
Feature Importance:



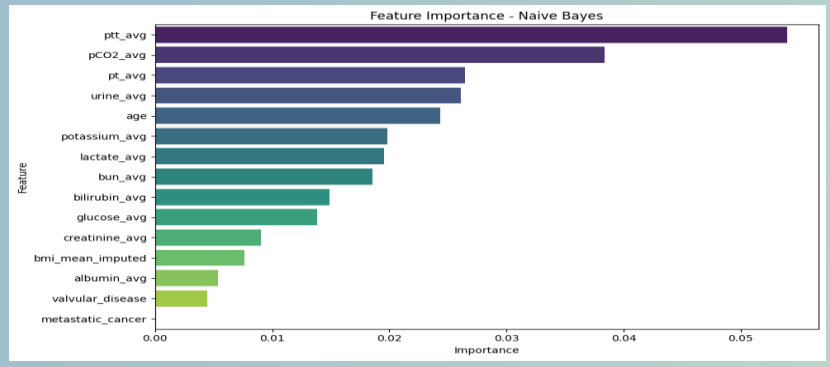
Random Forest



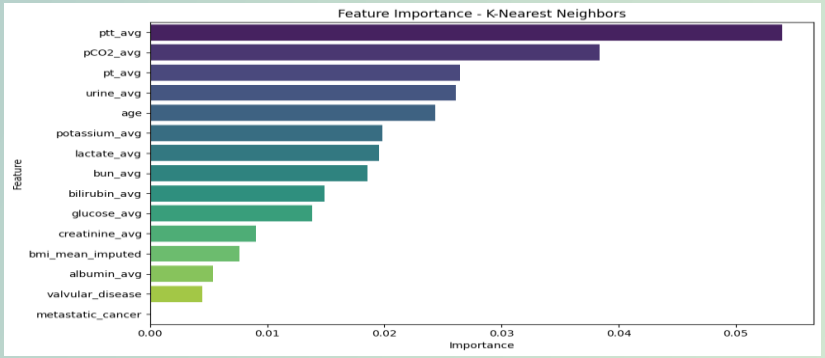
XG Boost



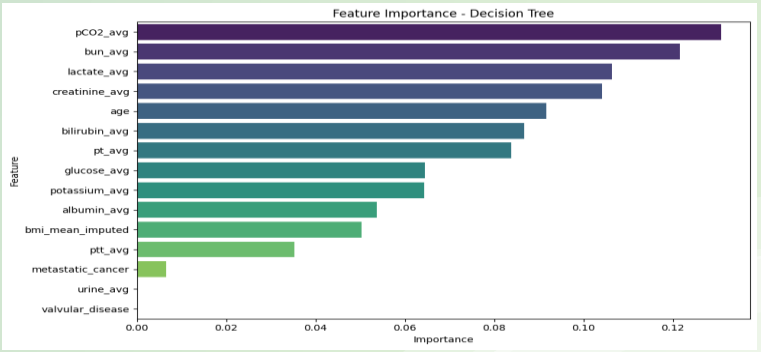
logistic Regression



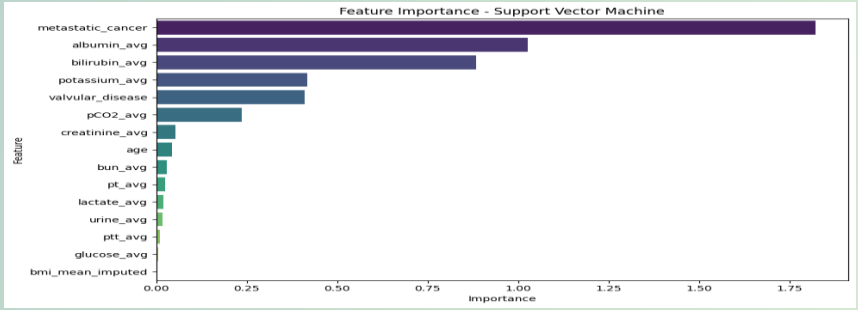
Navies Bayes



K-nearest neighbors



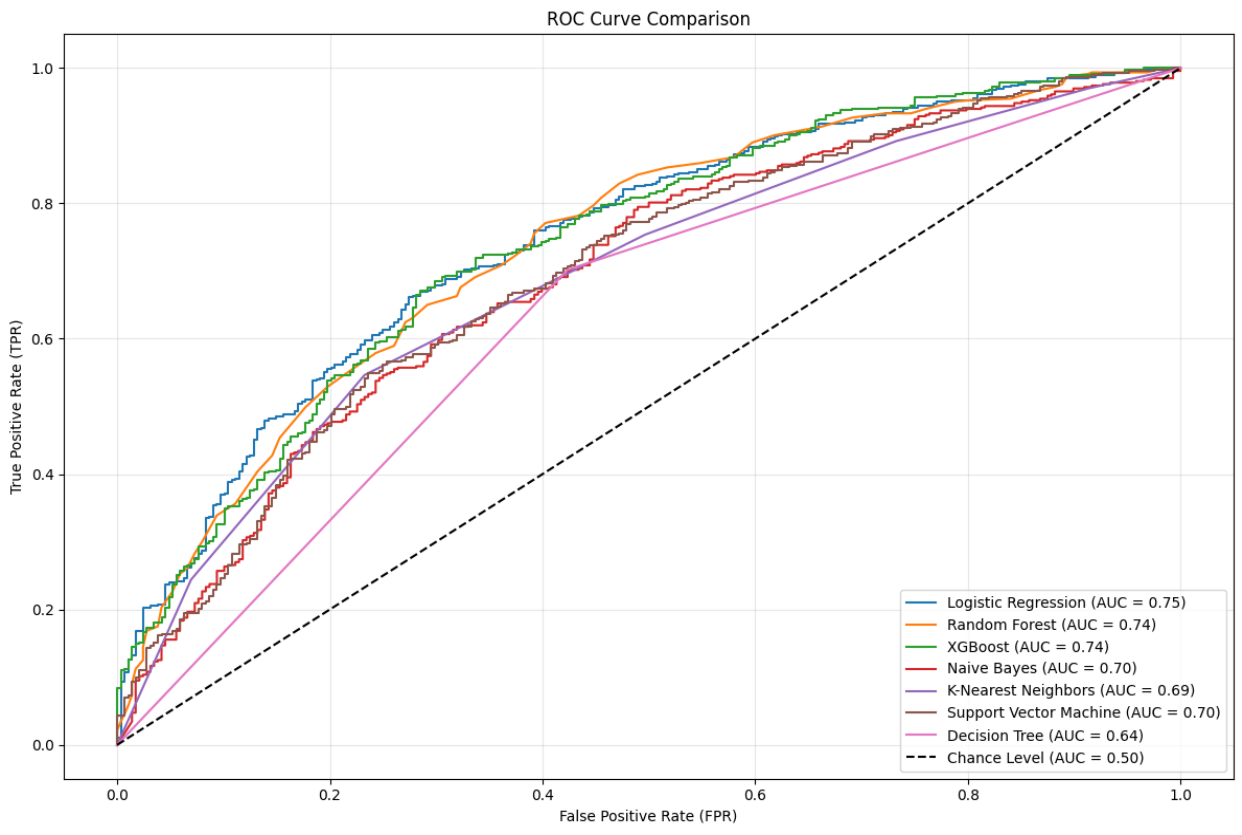
Decision Tree



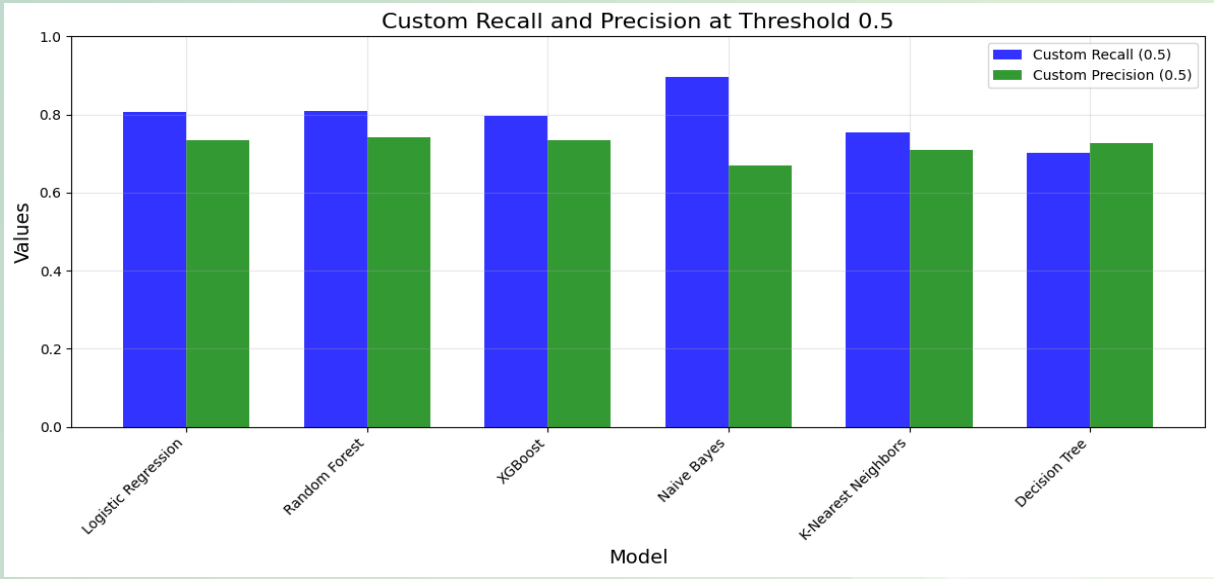
Support vector Machine

Model Validation:

- Logistic regression have highest AUC



Naïve Bayes have highest Recall Rate



Recall and Precision Summary:

	Model	Custom Recall (0.5)	Custom Precision (0.5)
0	Logistic Regression	0.805616	0.734252
1	Random Forest	0.807775	0.740594
2	XGBoost	0.796976	0.735060
3	Naive Bayes	0.896328	0.668277
4	K-Nearest Neighbors	0.753780	0.709350
5	Decision Tree	0.701944	0.727069

Conclusion:

- Logistic Regression demonstrated the highest AUC (overall performance).
- Naive Bayes achieved the highest Recall (Sensitivity) rate.

Clinical Importance:

- In ARDS, Recall (Sensitivity) is prioritized to ensure all potential cases are identified.
- Missing a case (false negative) can delay life-saving interventions, posing severe risks to patients.

Naive Bayes, is highly effective in maximizing Recall, making it a preferred choice in critical clinical settings.