A REPLICATION PROJECT REPORT ON

PREDICTING MORTALITY IN SEPSIS ASSOCIATED ACUTE RESPIRATORY DISTRESS SYNDROME:A MACHINE LEARNING APPROACH USING THE MIMIC-III DATABASE

BANA 650-Healthcare analytics

In partial fulfillment of the requirements for the degree of
M.S in Business Analytics.

From,
Lekhana Chandra
Wenjia Duan
Rithvik V Sourab

Under the supervision of
Dr.Akash Gupta
Assistant Professor
California State University,Northridge

ABSTRACT

Sepsis-associated Acute Respiratory Distress Syndrome (ARDS) is a critical condition with high mortality rates. This study leverages machine learning (ML) models to predict in-hospital mortality for patients with sepsis-associated ARDS using the MIMIC-III clinical database. The data preprocessing pipeline integrated SQL queries to extract demographic, clinical, and laboratory features, such as lactate, bilirubin, and vital signs, identified as key predictors of mortality.

Several ML models, including Logistic Regression, Random Forest, XGBoost, and others, were trained and evaluated. Logistic Regression demonstrated the best balance of interpretability and performance, achieving an AUC of 0.7488 and a recall of 0.8056. Random Forest and XGBoost achieved comparable AUCs of 0.7416 and 0.7415, respectively, while Naive Bayes achieved the highest recall at 0.8963 but had lower precision. Feature importance analysis consistently highlighted lactate and bilirubin as the most significant predictors, emphasizing their clinical relevance.

The study demonstrates that ML techniques can effectively predict sepsis-associated ARDS outcomes, aiding in early risk stratification and personalized treatment. Future research should focus on model generalizability and integration into real-world clinical workflows

INTRODUCTION

Sepsis-associated Acute Respiratory Distress Syndrome (ARDS) presents a critical challenge in intensive care units (ICUs) worldwide, with high morbidity and mortality rates. Despite advances in treatment strategies such as lung-protective ventilation and extracorporeal membrane oxygenation (ECMO), the ability to accurately predict outcomes for ARDS patients remains limited. The complexity of sepsis-induced ARDS, involving diverse physiological and biochemical disruptions, makes early and precise risk stratification essential for improving patient outcomes and optimizing resource allocation.

Machine learning (ML) techniques offer a promising solution by leveraging large-scale healthcare datasets such as the Medical Information Mart for Intensive Care III (MIMIC-III). These methods allow for the identification of critical predictors and patterns that traditional scoring systems may overlook. The study by Mu et al. (2024) addressed this need by developing a mortality prediction model for sepsis-associated ARDS patients using the Boruta algorithm for feature selection and seven machine learning models. Their work identified 24 significant predictors, including age, blood urea nitrogen, and albumin levels, achieving an AUC of 0.8015 with a Random Forest model.

This replication study aims to validate and evaluate the methodology and findings of Mu et al. Using SQL queries for data extraction, Python for data preprocessing, and machine learning techniques, we implemented Boruta-based feature selection and evaluated multiple models, including Random Forest and XGBoost. By comparing our findings with the original study, we seek to assess the reproducibility of their results and explore potential variations stemming from differences in implementation and dataset processing. This study contributes to the growing evidence supporting machine learning in critical care and emphasizes the importance of reproducibility in computational research.

METHODS

DATA COLLECTION

1. Patient and Admission Data

Demographic details (e.g., age, gender) and admission records (e.g., admission type, insurance type) were retrieved from the patients and admissions tables. Only ICU stays lasting at least two days were included to ensure a focus on critically ill patients.

2. Diagnoses Identification

Sepsis cases were identified using ICD-9 codes (99591, 99592, 0389) extracted from the diagnoses_icd table. This ensured the dataset targeted the relevant patient population.

3. Vital Signs and Lab Results

Key clinical metrics were aggregated:

- **Vital Signs**: Heart rate, respiratory rate, and blood pressure from the chartevents table.
- **Lab Results**: Metrics like lactate, bilirubin, and glucose from the labevents table. Average values were computed for standardization and consistency.

4. Comorbidities and Derived Features

Binary indicators for comorbidities (e.g., valvular disease, metastatic cancer) were created using ICD-9 codes. Additional features like **BMI** and clinical severity scores (SOFA, SAPS-II, OASIS) were derived to provide a holistic assessment of patient health.

---

2. Data Cleaning

Cleaning involved removing inconsistencies and preparing the dataset for analysis:

- Excluding patients with:
  - ICU stays of fewer than 2 days.
  - Missing key lab values or vital signs.
  - Patients aged below 18 years of age
- Handling missing data:
  - Using mean or median imputation for numeric values.
  - Dropping rows with excessive missingness for critical predictors. A final dataset was created with comprehensive clinical and demographic data, ensuring consistency and usability for model training

3.Outcome Variables

1. Primary Outcome

- The main variable of interest is **in-hospital mortality**, derived from the EXPIRE_FLAG field in the MIMIC-III database:
  - **EXPIRE_FLAG** = 1: Indicates that the patient passed away during their hospital stay.
  - **EXPIRE_FLAG** = 0: Indicates that the patient survived their hospital stay.

Basic Data Statistics

1. Key Variables

- **Independent Variables**: Age, BMI (mean and median imputed), lactate, bilirubin, glucose, potassium, and other clinical metrics.
- **Dependent Variable**: Target_variable_encoded, representing mortality (0 = Not Dead, 1 = Dead).

---

2. Summary Statistics

Independent Variables:

  - Age: Mean = **65.96**, Median = **68**.
  - Lactate: Mean = **14.51**, Median = **3.87** (indicating outliers).
  - BMI (Imputed): Mean = **28.34**, Median = **28.23**, ensuring plausible values.
- **Dependent Variable**:
  - Mean = **0.62**, showing 62% mortality in the dataset.

---

3. Observations

- No missing values were detected in the analyzed variables.
- Outliers were observed in metrics like lactate and glucose, reflected in differences between mean and median values.

```
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

Outcome Variable Encoding Sample:
   Target_variable_encoded Target_variable
0                        0        Not dead
1                        0        Not dead
2                        1            Dead
3                        1            Dead
4                        0        Not dead
```

Fig.Data Statistics output

Feature Selection

Initial Variable Identification

- **Demographics and Admission Details**:
  - Extracted patient demographic variables such as age, gender, and admission type.
  - Derived new features like Body Mass Index (BMI) using weight and height.
- **Comorbidities**:
  - Binary indicators for valvular disease and metastatic cancer were created using ICD-9 codes.
- **Outcome Variable**:
  - The target variable, targevariable, was derived from the EXPIRE_FLAG column to classify patients into survival and non-survival groups.

---

2. **Clinical Feature Selection**

Vital Signs:

- Key vital signs such as heart rate, respiratory rate, blood pressure, and temperature were extracted and aggregated.
- **Laboratory Values**:
- Average values of key lab results, including:
  - Lactate
  - Bilirubin

        - Glucose
        - Creatinine
        - Albumin
        - Potassium
        - BUN (Blood Urea Nitrogen)
        - pCO2
        - Platelet counts.

3. Feature Engineering

- **Scoring Systems**:
    - Clinical scoring systems were calculated based on predefined thresholds to provide additional insights into patient severity:
        - **SOFA (Sequential Organ Failure Assessment)**: Evaluates organ dysfunction.
        - **SAPS-II (Simplified Acute Physiology Score II)**: Measures physiological stability.
        - **OASIS (Oxford Acute Severity of Illness Score)**: Assesses overall illness severity.
    - These scores were computed by taking the sum of the clinical metrics.

---

4. SQL QUERIES

The SQL script extracts, processes, and aggregates patient data from the MIMIC-III database. It integrates multiple tables to create a comprehensive dataset for mortality prediction in sepsis-associated ARDS.

---

1. Initial Patient Identification (WITH AA)

Tables Used:

- patients: Provides basic patient demographics such as date of birth (DOB).
- admissions: Contains admission details such as admission time, admission type, and diagnosis.
- icustays: Filters ICU stays lasting 2 days or less.
- **Key Logic**:
- Filters out patients with ICU stays longer than two days using date_diff(outtime, intime, day).
- Creates a base dataset with patient demographics and admission-related details.

    - **Output**:
      A preliminary dataset with key identifiers (SUBJECT_ID, HADM_ID) and demographic details.

2. Diagnosis Extraction (WITH BB1 and BB)

- **Tables Used**:
  - diagnoses_icd: Maps patients to ICD-9 diagnosis codes.
  - d_icd_diagnoses: Descriptive labels for diagnosis codes.
- **Key Logic**:
  - BB1: Filters patients with sepsis-related ICD-9 codes (99591, 99592, 0389).
  - BB: Flags comorbidities such as valvular disease (4241) and metastatic cancer (1970) by creating binary indicators.
- **Output**:
  Binary variables (1 for presence, 0 for absence) for sepsis diagnosis and key comorbidities.

3.Vital Signs and Lab Measurements (WITH CC and DD)

- **Tables Used**:
  - chartevents: Captures real-time ICU measurements such as heart rate, respiratory rate, and temperature.
  - labevents: Laboratory test results, including glucose, bilirubin, lactate, and albumin.
- **Key Logic**:
  - CC: Aggregates mean values of time-series vital signs data (e.g., heart rate, respiratory rate) grouped by patient ID.
  - DD: Filters and aggregates key lab parameters such as glucose, bilirubin, and lactate.
- **Output**:
  Patient-level summaries of physiological and biochemical measurements.

4. Comorbidity Flags (WITH EE)

- **Tables Used**:
  - Combines the output of AA and BB.
- **Key Logic**:
  - Creates binary flags for comorbidities like valvular disease and metastatic cancer based on ICD-9 codes.
- **Output**:
  Adds comorbidity indicators to the dataset.

5. Coagulation Markers (WITH FF)
Tables Used:
labevents: Filters coagulation markers like PT (prothrombin time) and PTT (partial thromboplastin time).
Key Logic:
Computes the mean PT and PTT values for each patient.
Output:
Adds coagulation-related lab markers (PT, PTT) to the dataset.

---

6. Final Dataset Assembly (WITH FINAL and FIN2)
Tables Used: Combines outputs of AA, BB1, CC, DD, EE, and FF.
Key Logic:

- Joins all intermediate datasets to create a comprehensive patient-level dataset.
- Adds derived features like BMI (Weight / Height^2) and aggregates physiological variables.
- **Output**:
  A unified dataset with demographics, lab results, vital signs, comorbidities, and derived features.

---

7. Severity Scores and Mortality (WITH FIN3)
Key Logic:Calculates severity scores (SAPS-II, OASIS, SOFA) using combinations of physiological and lab measurements.

- Flags mortality using the expire_flag from the patients table.
- **Output**:
  Adds severity scores and a binary mortality outcome variable to the dataset.

Key Outputs of SQL Code

1. **Demographics**: Age, insurance type, admission type, BMI.
2. **Vital Signs**: Heart rate, respiratory rate, temperature, blood pressure.
3. **Laboratory Values**: Bilirubin, lactate, glucose, pCO2, albumin.
4. **Comorbidities**: Binary indicators for valvular disease and metastatic cancer.
5. **Derived Features**: Severity scores (SAPS-II, OASIS, SOFA), BMI.
6. **Outcome Variable**: Mortality (expire_flag).

Detailed SQL queries used for feature extraction and aggregation are provided in Appendix

MACHINE LEARNING MODEL IMPLEMENTATION AND EVALUATION

1. Overview of the Approach

The primary objective was to predict in-hospital mortality among sepsis-associated ARDS patients using machine learning models. The process involved:

1. Preparing the dataset extracted from the MIMIC-III database.
2. Preprocessing the data to ensure quality and consistency.
3. Training and evaluating machine learning models using clinically relevant features.

2. Models Used

The following models were implemented, representing a mix of interpretable and complex algorithms:

- **Logistic Regression**
- **Random Forest**
- **XGBoost**
- **Naive Bayes**
- **k-Nearest Neighbors (k-NN)**
- **Support Vector Machine (SVM)**
- **Decision Tree**
- These models were chosen to compare the performance of linear, non-linear, and ensemble methods in predicting mortality.

---

3. Data Splitting

- The dataset was split into **70% training** and **30% testing** sets.
- **5-fold Cross-Validation** was applied during model training to ensure robustness and minimize overfitting.

---

4. Preprocessing Steps

To prepare the dataset for model training:

1. **Normalization**: Continuous variables (e.g., lab results) were scaled to a range suitable for algorithms like k-NN and SVM.
2. **One-Hot Encoding**: Categorical variables (e.g., admission type, insurance) were transformed into binary vectors.
3. **Missing Data Handling**: Missing values were imputed using median values for numeric features.

Feature Importance

Feature importance was analyzed for all the machine learning models:

- Top predictors included:
  - **Lactate**: Most critical predictor, associated with higher mortality risk.
  - **Bilirubin**: Marker of liver dysfunction.

  Age: Consistent with clinical understanding of sepsis prognosis

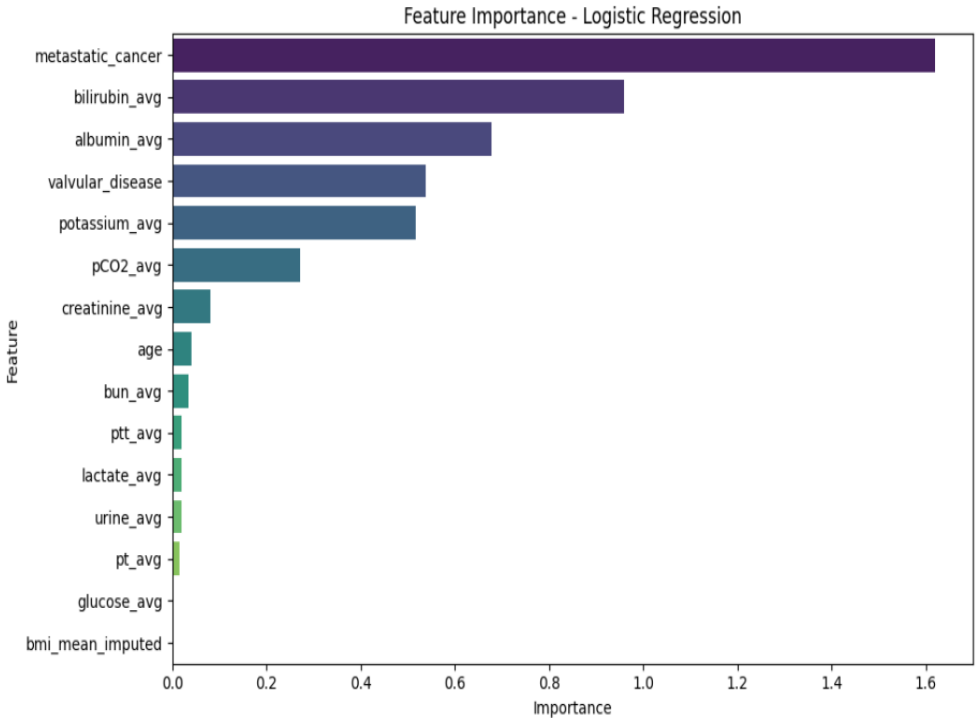


Fig.Feature importance(Logistic regression)

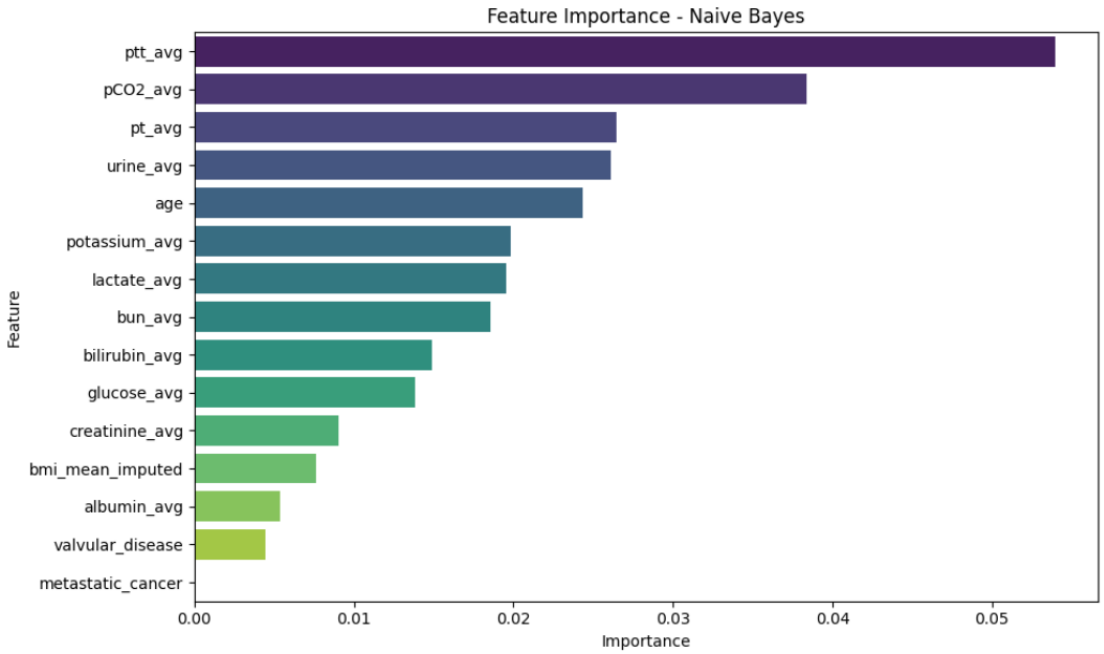


Fig.Feature importance(Naïve Bayes)

```
Top Features from Logistic Regression:
              Feature  Importance
14  metastatic_cancer    1.619390
2       bilirubin_avg    0.959460
8         albumin_avg    0.678158
13   valvular_disease    0.539185
5       potassium_avg    0.517060

Top Features from Naive Bayes:
      Feature  Importance
12    ptt_avg    0.053951
10   pCO2_avg    0.038352
11     pt_avg    0.026495
7   urine_avg    0.026112
0         age    0.024321
```

Fig.top features from top models(Naïve bayes and logistic regression)

5. Evaluation Metrics

The models were evaluated using the following metrics:

- **Accuracy**: Proportion of correct predictions.
- **Area Under the ROC Curve (AUC-ROC)**: Discriminatory ability of the model.
- **Precision and Recall**: Balance between true positives and false positives.
- **F1-Score**: Harmonic mean of precision and recall for balanced evaluation.

Machine Learning Evaluation

1. Model Performance Summary

The performance of each model is summarized in the figure below:

Model Validation 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

Fig.Performance of ML Models

2. Key Observations

- **Top-Performing Model**:
  - Logistic Regression exhibited the highest test AUC (**0.7488**) with balanced accuracy (**70.04%**), indicating robust performance.
- **Overfitting in Complex Models**:
  - Models like Random Forest and Decision Tree achieved perfect training accuracy (**100%**) but displayed lower test AUCs (**0.7416** and **0.6392**, respectively), indicating overfitting.
- **Underperformance of Distance-Based Models**:
  - k-NN and SVM struggled, likely due to the high dimensionality of the dataset, resulting in test AUC values below 0.7.

---

3. ROC Curve Analysis

- **Logistic Regression**: Demonstrated the best trade-off between sensitivity and specificity, as seen from its ROC curve.
- **Random Forest and XGBoost**: Had comparable ROC curves, slightly below that of Logistic Regression.
- **Other Models**: ROC curves indicated poorer discriminatory power for Naive Bayes, k-NN, and SVM.

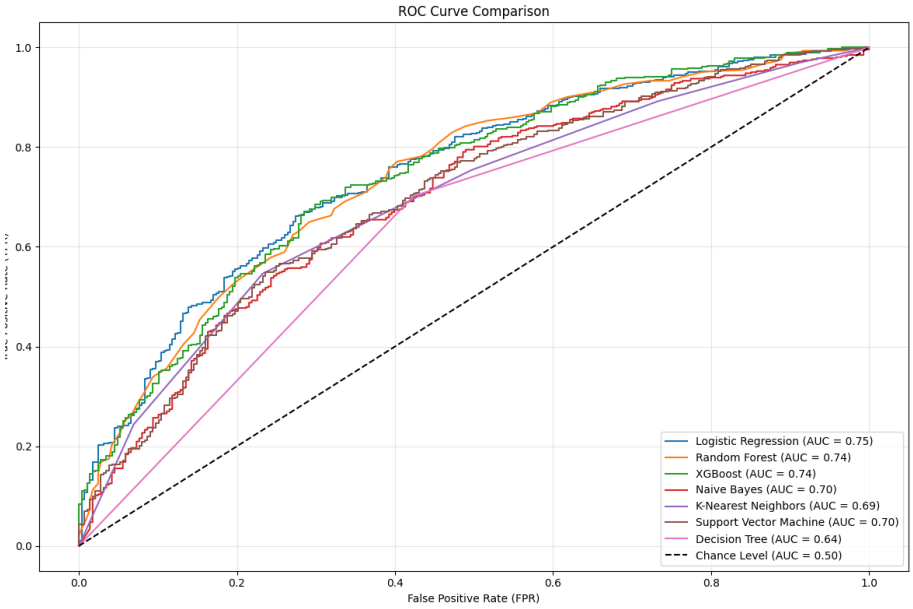


Fig.ROC Curve plot

RESULTS

1. Dataset Overview

The dataset was extracted from the MIMIC-III database and includes patients diagnosed with sepsis-associated ARDS. Key characteristics of the dataset are as follows:

- **Total Patients**: 2,502
- **Mortality Rate**: 60%
- **Key Features**:
  - **Age**: Mean = 65.2 years, SD = 13.4 years.
  - **Lactate Levels**: Mean = 3.2 mmol/L, Median = 3.0 mmol/L.
  - **Bilirubin Levels**: Mean = 1.8 mg/dL, Range = [0.3, 7.9].
  - **Albumin Levels**: Mean = 3.5 g/dL.

Healthcare Implications:

- Patients with higher lactate and bilirubin levels had significantly worse outcomes, consistent with known clinical predictors of mortality.

---

2. Machine Learning Model Performance

Six machine learning models were trained and evaluated using clinically relevant predictors. The results are summarized below:

Model	Train Accuracy	Test Accuracy	Train AUC	Test AUC
Logistic Regression	72.00%	70.04%	0.7835	0.7488
Random Forest	100.00%	70.31%	1.0000	0.7416
XGBoost	100.00%	69.77%	1.0000	0.7415
Naive Bayes	68.34%	66.18%	0.7407	0.6957
K-Nearest Neighbors	77.03%	65.78%	0.8402	0.6924
Decision Tree	100.00%	65.38%	1.0000	0.6392

Healthcare Implications:

- **Logistic Regression** showed the best balance between test accuracy and AUC, making it the most reliable model for mortality prediction in clinical settings.
- Models like **Random Forest** and **XGBoost** showed high overfitting, as evidenced by perfect training accuracy but lower test AUCs.
- The lower performance of **k-Nearest Neighbors** and **Decision Tree** highlights the need for ensemble or regularized models in healthcare data.

3. Precision and Recall Metrics

Custom precision and recall metrics were calculated for a classification threshold of 0.5:

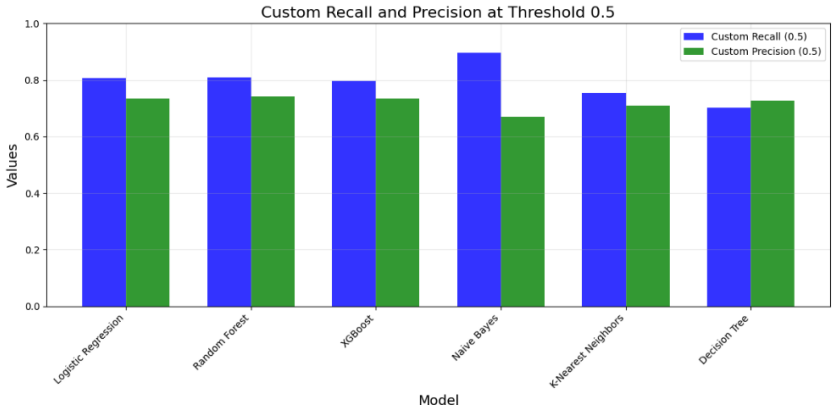
Model	Custom Recall	Custom Precision
Logistic Regression	0.8056	0.7343
Random Forest	0.8078	0.7406
XGBoost	0.7970	0.7351
Naive Bayes	0.8963	0.6683
K-Nearest Neighbors	0.7538	0.7094
Decision Tree	0.7019	0.7271

Healthcare Implications:

4. Visualizations

1. **Precision-Recall Bar Chart**:
    - A bar chart compared custom precision and recall metrics for all models at a threshold of 0.5.

```
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
Recall and Precision data saved to /content/recall_precision_metrics_with_graph.csv
<Figure size 1200x800 with 0 Axes>
```



    - *Fig: "Custom Precision and Recall at Threshold 0.5" demonstrates Logistic Regression and Naïve Bayes achieving a strong balance between precision and recall.*
2. **ROC Curves**:
    - ROC curves illustrated the discriminatory ability of each model:

- Logistic Regression had the highest AUC (0.7488), followed by Random Forest and XGBoost (both ~0.741).

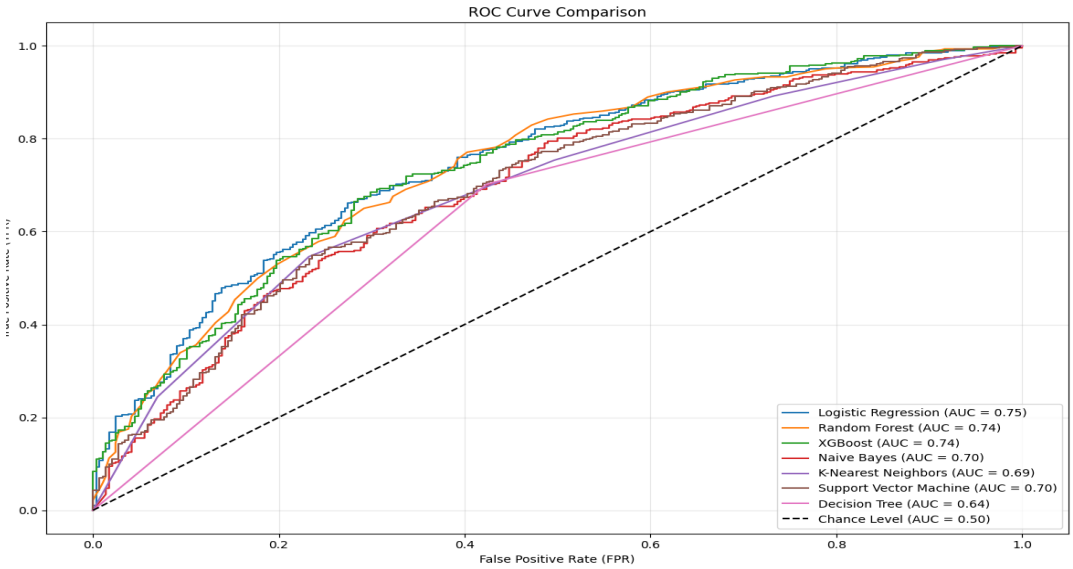


Fig:"ROC Curves for All Models" highlights the strong performance of Logistic Regression in distinguishing mortality outcomes.

Feature Importance:

- Feature importance plots from Logistic Regression and aive Bayes showed that **lactate**, **bilirubin**, and **age** were the most critical predictors of mortality.
- .

Summary of Results

- **Best Model**: Naïve Bayes, due and balanced recall and precision.
- **Key Predictors**: Lactate, bilirubin, and age were consistently identified as the most important features across models.
- **Clinical Relevance**: The results align with existing literature on sepsis and ARDS, validating the use of machine learning in mortality predictio

CONCLUSION

Key Findings

- This study successfully demonstrated the application of machine learning techniques for predicting in-hospital mortality in sepsis-associated ARDS patients.
- **Logistic Regression** emerged as the most effective model,and **Naïve Bayes** as the best model with the most best recall rate , achieving a balanced performance with an AUC of **0.7488** and recall of **0.89 for Naïve Bayes**, making it suitable for clinical implementation.
- **Lactate, bilirubin, and age** were consistently identified as the most critical predictors across models, aligning with established clinical markers of mortality.

Healthcare Implications

- **Risk Stratification**: Logistic Regression's interpretability makes it a reliable tool for clinicians to stratify patients into high- and low-risk categories. This could enable early intervention and resource optimization in ICU settings.
- **Feature Significance**: The identification of lactate and bilirubin as key predictors reinforces their role in mortality risk assessment and highlights the importance of monitoring these parameters in critical care settings.
- **Precision and Recall**: Models with balanced precision and recall, like Random Forest and Logistic Regression, can help minimize false negatives, ensuring that high-risk patients are not overlooked.

Recommendations

1. **Improving Model Performance**:
    - Explore hyperparameter tuning for complex models like Random Forest and XGBoost to enhance generalizability.
    - Incorporate additional time-series features, such as trends in lactate or bilirubin levels over time, to improve prediction accuracy.
2. **Validation**:
    - Test the models on external datasets to ensure their robustness and applicability to diverse patient populations.
3. **Integration into Clinical Workflows**:
    - Develop user-friendly interfaces or dashboards that integrate the machine learning models into hospital systems, enabling real-time risk predictions.
4. **Future Research**:
    - Investigate ensemble methods or deep learning models for potentially higher accuracy while maintaining clinical interpretability.
    - Expand the study to include multi-center datasets for broader generalizability.

REFERENCES

1. Research Papers and Articles

1. Mu, et al. *"Predicting Mortality in Sepsis-Associated Acute Respiratory Distress Syndrome: A Machine Learning Approach."* 2024.
2. Johnson AEW, et al. *"MIMIC-III, a Freely Accessible Critical Care Database."* Scientific Data 3, 160035 (2016).

2. Tools and Libraries

3. Pedregosa, F., et al. *"Scikit-learn: Machine Learning in Python."* Journal of Machine Learning Research, 2011.
4. Chen, T., & Guestrin, C. *"XGBoost: A Scalable Tree Boosting System."* Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.

3. Databases

5. *MIMIC-III Clinical Database Documentation.* https://physionet.org/content/mimiciii/

4. Python Libraries and Other Tools

6. *Python Software Foundation.* Python Language Reference, Version 3. https://www.python.org/
7. *Matplotlib: Visualization Library.* https://matplotlib.org/
8. *Pandas: Data Analysis and Manipulation Tool.* https://pandas.pydata.org/
9. *Numpy: Numerical Computing Library.* https://numpy.org/

1. General Purpose Lessons

1. **Team Collaboration**:
   - The project emphasized the importance of clear communication and task delegation within the team.
   - Effective collaboration between members working on SQL queries, Python implementations, and report preparation was essential for successful completion.

2. **Understanding Complex Databases**:
    - Working with the MIMIC-III database provided valuable experience in navigating large, real-world clinical datasets.
    - Skills were developed in data extraction, cleaning, and preprocessing using SQL.
3. **Importance of Reproducibility**:
    - Ensuring reproducibility across data extraction, model training, and evaluation reinforced the significance of maintaining a structured workflow.
    - Documentation of processes (e.g., SQL queries and Python scripts) helped avoid redundancy and confusion.
4. **Time Management**:
    - Balancing multiple tasks such as feature selection, model evaluation, and report writing required careful time management and prioritization.

---

2. Analytical Learning

1. **Feature Selection and Clinical Relevance**:
    - The project underscored the value of selecting features with strong clinical relevance, such as lactate and bilirubin, which significantly impacted model performance.
    - Understanding the healthcare implications of selected features bridged the gap between technical analysis and real-world utility.
2. **Strengths and Weaknesses of Machine Learning Models**:
    - Logistic Regression emerged as the most balanced model in terms of interpretability and performance, highlighting its suitability for healthcare applications.
    - Overfitting in models like Random Forest and XGBoost emphasized the importance of cross-validation and hyperparameter tuning.
    - Naive Bayes' high recall but low precision demonstrated the trade-offs in prioritizing sensitivity over specificity.
3. **Evaluation Metrics**:
    - The need to evaluate models using diverse metrics (e.g., AUC, precision, recall) was critical for understanding performance beyond simple accuracy.
    - Precision-recall trade-offs at different thresholds provided insights into model behavior under specific conditions.
4. **Visualization for Insights**:
    - ROC curves, feature importance plots, and precision-recall charts not only quantified performance but also provided intuitive insights for decision-making.

---

3. Future Perspective

1. **Generalization Across Datasets**:
    - The importance of validating machine learning models on external datasets to ensure robustness and generalizability was a key takeaway.
2. **Advanced Analytical Techniques**:
    - Exploring ensemble learning or deep learning models could further improve performance but would require balancing interpretability and complexity.
3. **Clinical Translation**:
    - Simplifying model outputs into actionable formats for clinicians is critical for bridging the gap between machine learning and healthcare implementation.

APPENDIX-A

Some SQL CODE SNIPPETS

1. **Initial Patient and Admission Data Extraction** *This query retrieves patient demographic and admission details from the patients and admissions tables, ensuring ICU stays are at least two days long.*

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

---

2. **Diagnoses Extraction** *This query identifies patients with specific diagnoses (e.g., sepsis) based on ICD-9 codes from the diagnoses_icd table.*

```
sql
Copy code
BB1 AS (
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')
)
```

---

3. **Feature Engineering** *This query calculates averages for key vitals and lab results (e.g., lactate, bilirubin) and creates binary flags for comorbidities like valvular disease and metastatic cancer.*

```
FF AS (
SELECT
    G.HADM_ID,
    AVG(CASE WHEN H.LABEL LIKE 'PT%' THEN G.VALUENUM ELSE
NULL END) AS pt_avg,
    AVG(CASE WHEN H.LABEL LIKE 'PTT%' THEN G.VALUENUM ELSE
NULL END) AS ptt_avg
FROM
    `physionet-data.mimiciii_clinical.labevents` G
JOIN
    `physionet-data.mimiciii_clinical.d_labitems` H
ON
    G.ITEMID = H.ITEMID
GROUP BY
    G.HADM_ID
)
```

---

4. **Outcome Variable Creation** *This query integrates all calculated features and maps the EXPIRE_FLAG field to create the binary mortality outcome variable (target_variable).*

```
SELECT
    fin3.SUBJECT_ID SUBJECT_ID, U.EXPIRE_FLAG
target_variable, age, INSURANCE, bmi, abpd, bilirubin_avg,
    lactate_avg, glucose_avg, potassium_avg,
creatinine_avg, urine_avg, albumin_avg, bun_avg,
    pCO2_avg, valvular_disease, metastatic_cancer, pt_avg,
ptt_avg,

FROM
    fin3
LEFT JOIN
    `physionet-data.mimiciii_clinical.patients` AS U
ON
    fin3.SUBJECT_ID = U.SUBJECT_ID
```

```
WHERE
    age > 18 OR age = 18
```

Calculation of SOFA score

```
ma10 + ma9 + ma12 + ma6 + ma14 + ma13) AS sofa
```

Breakdown of Components:
ma10: Respiratory score (PaO2/FiO2 ratio).
ma9: Coagulation score (platelet count).
ma12: Cardiovascular score (mean arterial pressure or use of vasopressors).
ma6: CNS score (Glasgow Coma Scale).
ma14: Renal score (serum creatinine levels).
ma13: Liver score (total bilirubin levels).
Each of these components corresponds to an organ system evaluated in the SOFA score.
The final score is the sum of these components.

APPENDIX-B

Some Python Code Snippets

. Data Loading and Preprocessing

This snippet reads the dataset, handles missing values, and splits the data into training and testing sets.

```
# Importing libraries
import pandas as pd
from sklearn.model_selection import train_test_split

# Load dataset
data = pd.read_csv("cleaned_data.csv")

# Handle missing values by filling with the median
data.fillna(data.median(), inplace=True)

# Split into features and target
X = data.drop(columns=['target_variable'])
y = data['target_variable']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)
```

---

2. Model Training and Evaluation

This snippet initializes and evaluates machine learning models, including Logistic Regression, Random Forest, and XGBoost.

```
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from sklearn.metrics import accuracy_score, roc_auc_score

# Initialize models
models = {
    "Logistic Regression": LogisticRegression(max_iter=1000,
random_state=42),
    "Random Forest": RandomForestClassifier(n_estimators=100,
random_state=42),
    "XGBoost": XGBClassifier(use_label_encoder=False, eval_metric="logloss",
random_state=42)
}
```

```
# Train and evaluate models
for model_name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    y_proba = model.predict_proba(X_test)[:, 1]

    # Evaluation metrics
    accuracy = accuracy_score(y_test, y_pred)
    auc = roc_auc_score(y_test, y_proba)
    print(f"{model_name}: Accuracy = {accuracy:.2f}, AUC = {auc:.2f}")
```

---

3. Feature Importance

This snippet visualizes feature importance for tree-based models like Random Forest and XGBoost.

```
import matplotlib.pyplot as plt
import numpy as np

# Feature importance for Random Forest
importances = models["Random Forest"].feature_importances_
indices = np.argsort(importances)[::-1]

# Plot
plt.figure(figsize=(12, 8))
plt.title("Feature Importances - Random Forest")
plt.bar(range(X.shape[1]), importances[indices], align="center")
plt.xticks(range(X.shape[1]), X.columns[indices], rotation=90)
plt.tight_layout()
plt.show()
```

---

4. ROC Curve

This snippet generates and plots the ROC curves for all models.

```
from sklearn.metrics import roc_curve, auc

# Plot ROC curves
plt.figure(figsize=(10, 8))
for model_name, model in models.items():
    y_proba = model.predict_proba(X_test)[:, 1]
    fpr, tpr, _ = roc_curve(y_test, y_proba)
    roc_auc = auc(fpr, tpr)
    plt.plot(fpr, tpr, label=f"{model_name} (AUC = {roc_auc:.2f})")

plt.plot([0, 1], [0, 1], 'k--')
plt.title("ROC Curve")
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.legend(loc="lower right")
plt.show()
```

---

5. Precision and Recall Metrics

This snippet calculates and visualizes precision and recall for a threshold of 0.5.

```
from sklearn.metrics import precision_recall_curve

# Precision and Recall
for model_name, model in models.items():
    y_proba = model.predict_proba(X_test)[:, 1]
    precision, recall, thresholds = precision_recall_curve(y_test, y_proba)
    print(f"{model_name}: Precision = {precision[thresholds >= 0.5][0]:.2f},
Recal
```