Leveraging Business Analytics to Tackle Healthcare Challenges: Predicting Mortality with Data-Driven Insights

1. Data Exploration and Preprocessing

1.1. Dataset Combination

We focus on seven datasets from MIMIC III and select crucial features to process the final dataset for mortality prediction. To accurately reflect the required events, we organized the data into three intermediate tables: User Information Table(patient demographic data, hospital admission records, and ICU stay information); Procedure and Events Table(focuses on ICU procedures and events, such as invasive and non-invasive ventilatory support); Input Measurements Table(input measurements, such as fluids, medications).

1.1.1. Intermediate datasets combination

- **Dataset_1: User Information Table**

PATIENTS Table: The patients table uses subject_id as the primary key, uniquely identifying each patient.
ADMISSIONS Table: The admissions table uses hadm_id as the primary key, representing a unique hospital admission for a patient. Since a patient can have multiple admissions, a left join is performed between patients and admissions on the subject_id column to link each patient to their respective admissions.
ICUSTAYS Table: The icustays table uses icustay_id as the primary key, representing unique ICU stays. A single hospital admission (hadm_id) may have zero, one, or multiple ICU stays. A left join is performed between the admissions and icustays tables on the hadm_id column to associate admission records with their respective ICU stays.
Resulting Intermediate Table: The above joins produce an intermediate table that consolidates patient demographics, hospital admission details, and ICU stay information.

This process effectively integrates patient, admission, and ICU data for further analysis.

- **Dataset_2: Procedure and Events Table**:

To accurately reflect the severity of critical illness, we extracted the **mechanical ventilation** status as a binary feature (mechanical_ventilation) for each ICU stay. This variable indicates whether a patient received **invasive or non-invasive ventilatory**

support within the **first 24 hours** of ICU admission—a strong clinical marker associated with increased mortality risk.

Given the structure of the MIMIC-III database, we leveraged two key sources:

- **PROCEDUREEVENTS_MV.csv**: For MetaVision ICU stays, we directly identified intubation and ventilatory procedures using a curated list of itemids (e.g., *Invasive Ventilation*, *Non-invasive Ventilation*).

- **CHARTEVENTS.csv**: For CareVue and MetaVision systems, we inferred mechanical ventilation indirectly based on respiratory-related charted variables (e.g., *FiO2*, *PEEP*, *Ventilation Mode*) using a clinically verified itemid set.

We then combined both sources by checking if any ventilation-related entries occurred within **24 hours after ICU admission (intime)**. If so, we assigned mechanical_ventilation = 1; otherwise, it was set to 0.

- **Dataset_3: Input Measurements Table**

This process involves selecting relevant columns from two datasets (INPUTEVENTS_CV and INPUTEVENTS_MV), then identifying vasopressors (such as Epinephrine, Dopamine, and Vasopressin) by matching ITEMIDs with a predefined list of medication codes. A new column named VASOPRESSOR_USE is added to indicate whether each record corresponds to a vasopressor (1 = yes, 0 = no). The key information from INPUTEVENTS_CV and INPUTEVENTS_MV is then combined into a single DataFrame. The data is grouped by patient identifiers (SUBJECT_ID, HADM_ID, ICUSTAY_ID) to determine whether any vasopressors were used during each ICU stay. Finally, these unique identifier combinations are merged with the vasopressor usage data to produce a final result indicating whether vasopressors were administered during each patient stay.

1.1.2 Final Dataset

Finally, we used MySQL Workbench to join the three intermediate tables (dataset_1, dataset_2, and dataset_3) into a final table by connecting them through their reference keys. This final table was then imported into Python for further preprocessing and analysis.

To prepare the final dataset for prediction, we followed these steps:

We used subject_id as the primary key to structure the final table, consolidating data into a funnel table. This table recorded each user's counts of hospital admissions and ICU stays, providing an overview of their medical journey.

For admission information, events, and input measurements, we retained only the most recent records for each user. This ensured that the dataset focused on the latest and most relevant medical data for prediction tasks.

Additionally, since the timestamps for birth date, admission date, and ICU stays in the medical data were intentionally perturbed with noise, we calculated the key age metrics for each patient at critical points. By subtracting the respective timestamps, we derived users' ages at hospital admission and during ICU stays.

The result was a comprehensive patient summary report, reflecting each user's overall visit patterns, key medical milestones, and the most recent data for prediction.

1.2 Exploratory Data Analysis (EDA)

Q1. Data description

o Summary Statistics:

Numerical Variables:

	admission_count	adm_age	los_adm	icu_count	icu_age	los_icu	mechanical_ventilation	VASOPRESSOR_USE	death_age	hospital_expired
count	46520.0	46520.0	46520.0	46520.0	46467.0	46467.0	46467.0	46467.0	5813.0	46520.0
mean	1.3526	62.1831	9.8755	1.3268	62.1856	4.9958	0.8795	0.2197	89.4193	0.1249
std	1.0262	57.5430	12.4667	1.0038	57.5548	10.2160	0.3255	0.4141	70.0113	0.3306
min	1.0	0.0	-0.9451	0.0	0.0	0.0002	0.0	0.0	0.0	0.0
25%	1.0	38.8	3.5743	1.0	38.8	1.0667	1.0	0.0	59.8	0.0
50%	1.0	60.9	6.184	1.0	60.9	2.0235	1.0	0.0	73.5	0.0
75%	1.0	75.9	11.3472	1.0	75.9	4.3779	1.0	0.0	82.9	0.0
max	43.0	311.6	206.4257	42.0	311.6	171.6227	1.0	1.0	310.2	1.0

- **Age-related features(admission_age, icu_age, death_age):** admission_age and icu_age have a mean around 62 years, death_age has a mean around 89.4 years, though some values exceed 300 years, which may indicate data anomalies.

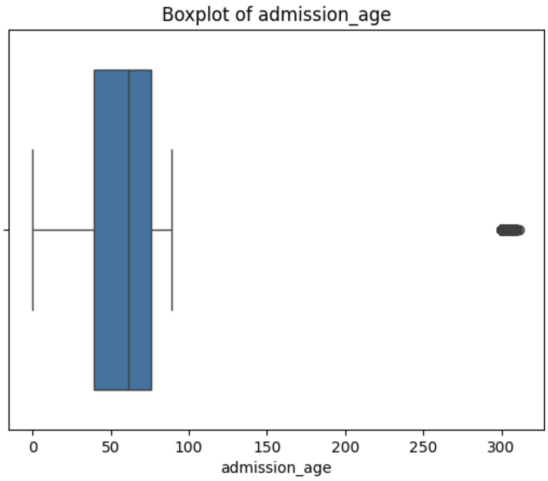
- **admission_count & icu_count:** 75% percentile = 1 means that at least 75% of the patients only have 1 admission or 1 ICU stay; However, maximum value is 43, which is drastically higher than the 75th percentile. This may indicate extreme outliers.
- **los_adm & los_icu :** For los_adm, there are negative values (as low as -0.94), which are logically invalid since patients cannot be discharged before admission—these likely stem from data entry or timestamp errors; The maximum value of 206.43 days is an extreme outlier, far exceeding the typical range (with 75% of stays under 11.35 days). Similarly, los_icu has a maximum value of 171.62.

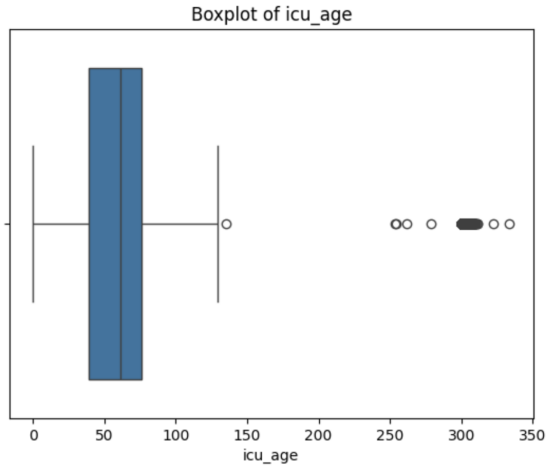
Categorical Variables:

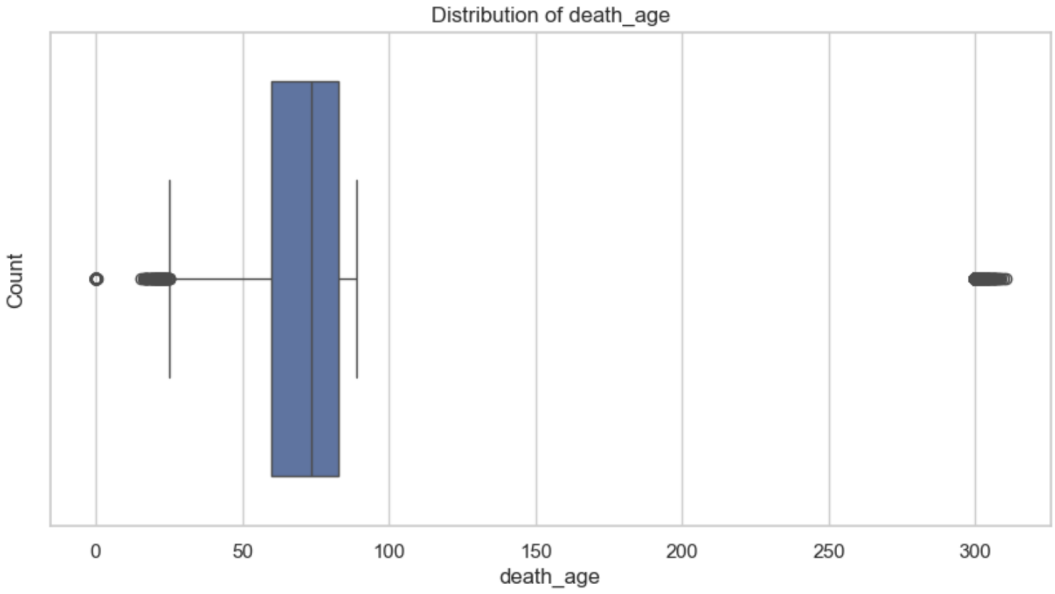
The dataset contains 8 categorical features, including key demographic and administrative attributes such as gender, insurance, language, and religion.

Categorical Variable	Unique Values
gender	2
insurance	5
language	75
religion	20
marital_status	7
ethnicity	41
last_admission_location	9
last_admission_type	4

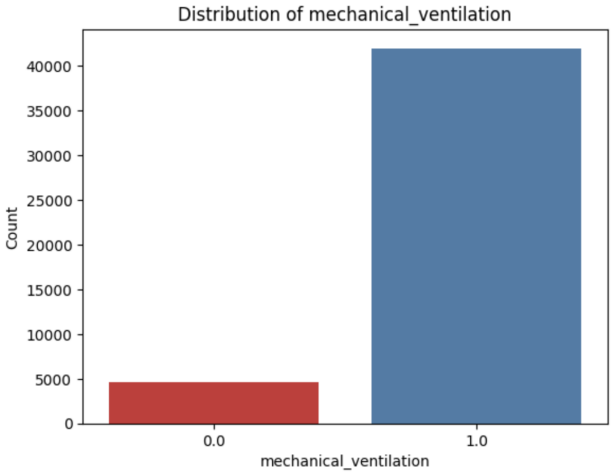
- **Univariate Analysis:**

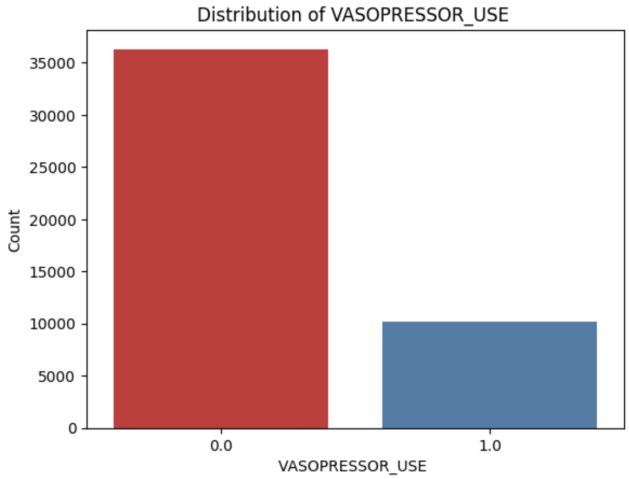




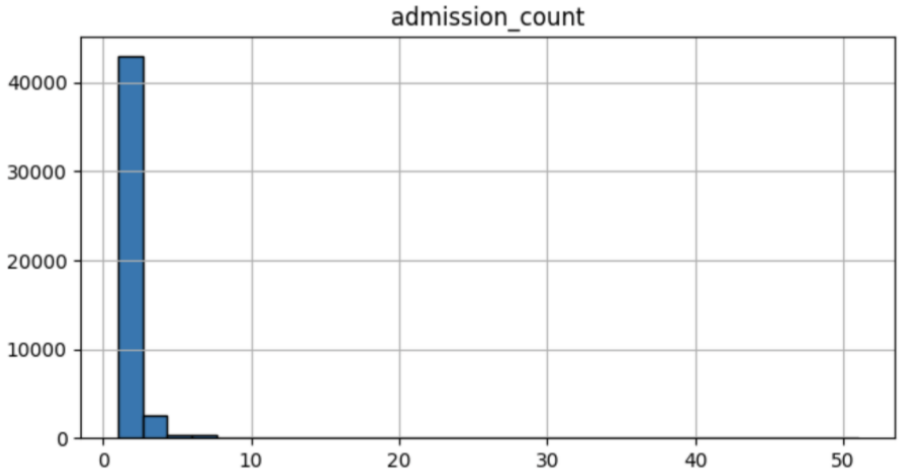


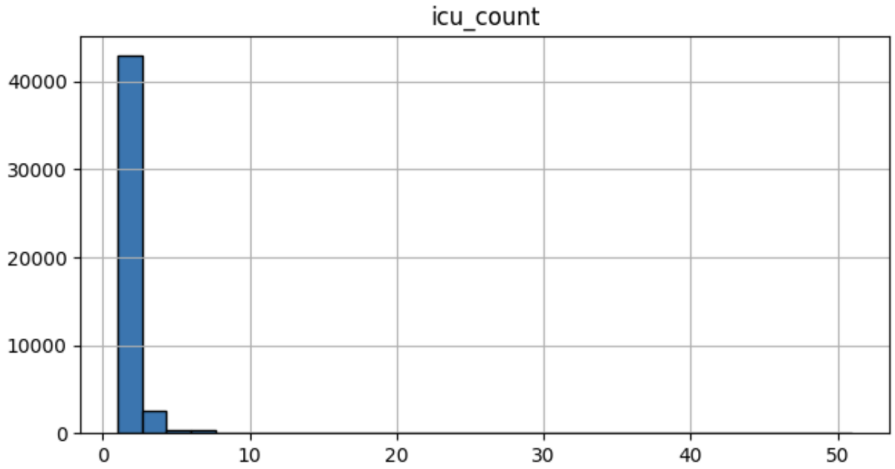
The boxplots for admission_age, icu_age, and death_age all reveal the presence of extreme outliers, with several values exceeding 300 years—well beyond biological plausibility. These outliers are likely due to data entry errors and should be carefully handled through filtering or capping to ensure data quality and prevent skewed model outcomes.



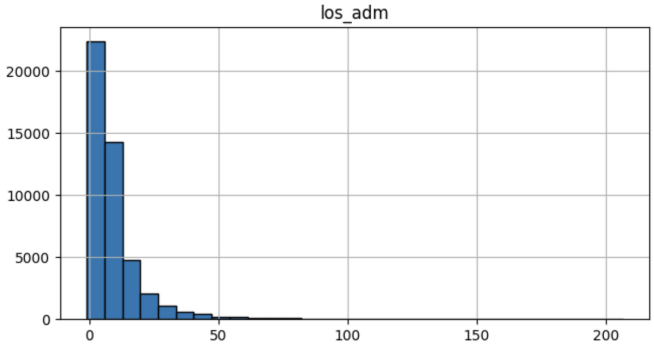


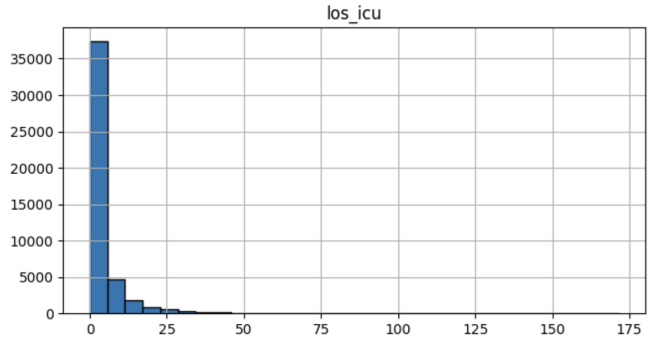
The majority of patients in the dataset received mechanical ventilation, while the use of vasopressors was less common. This suggests that mechanical ventilation was a standard intervention in critical care, whereas vasopressors were likely reserved for more severe or specific conditions.



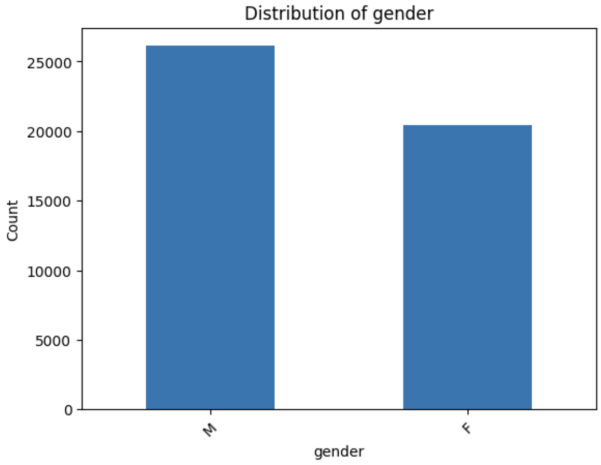


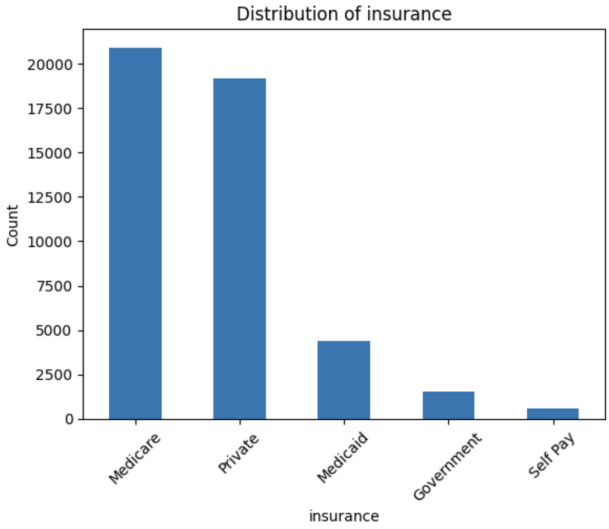
Both admission_count and icu_count are highly right-skewed, with the vast majority of patients experiencing only one hospital admission and ICU stay. However, a small number of patients show significantly higher counts—up to 40 or more—indicating extreme outliers or rare chronic cases.





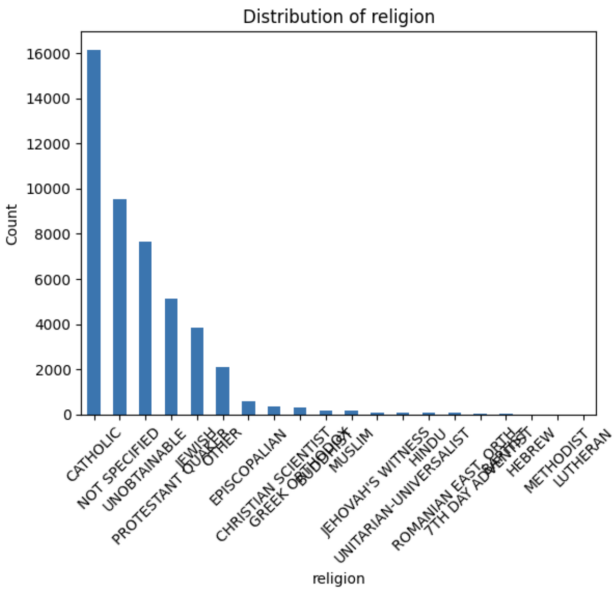
Both los_adm (length of hospital stay) and los_icu (length of ICU stay) show strong right-skewed distributions, with most patients staying for only a few days. However, there are noticeable long-tail outliers, including hospital stays over 200 days and ICU stays exceeding 170 days.

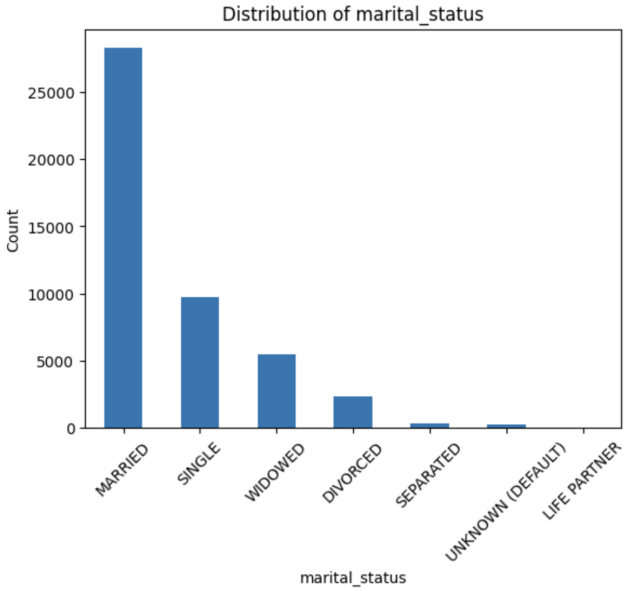




Male patients are slightly more common than female patients.

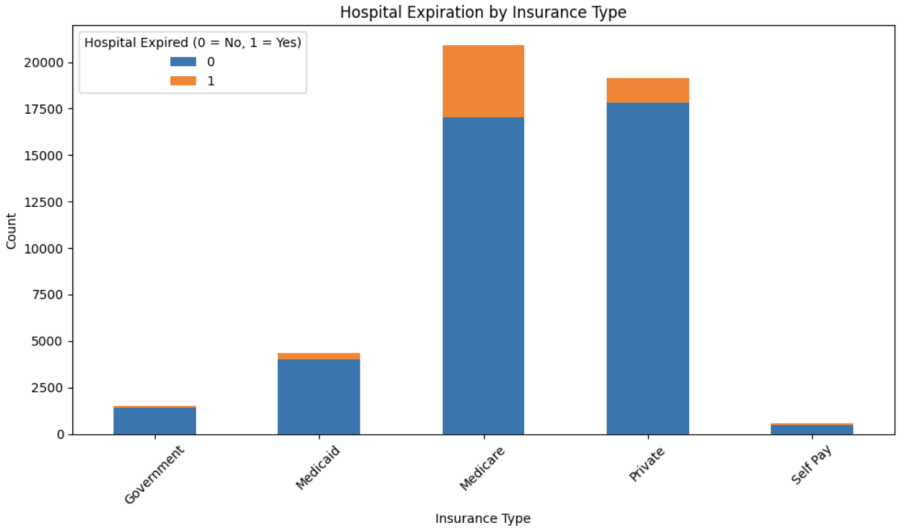
Medicare and private insurance are the dominant payer types, while options like self-pay and government programs are rare.



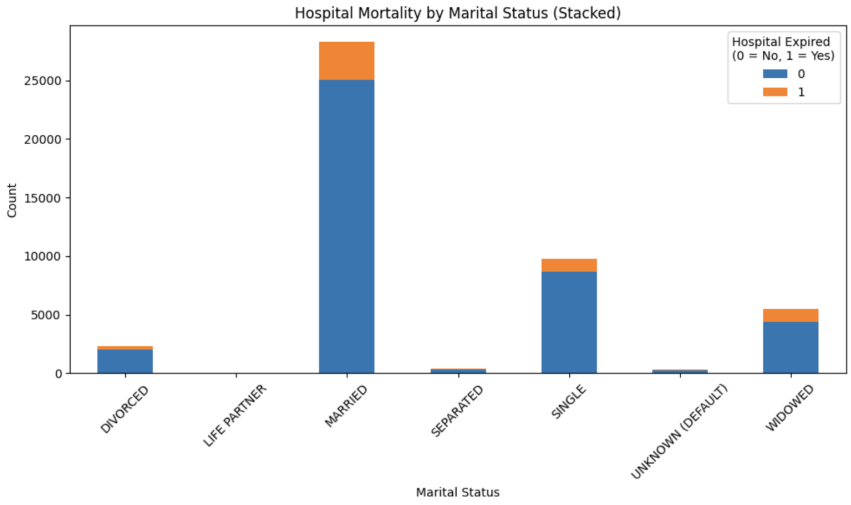


In terms of religion, Catholicism is the most frequently recorded affiliation, though a large portion is also unspecified or unobtainable. For marital status, the majority of patients are married, followed by single and widowed, with very few listed as separated or life partners.

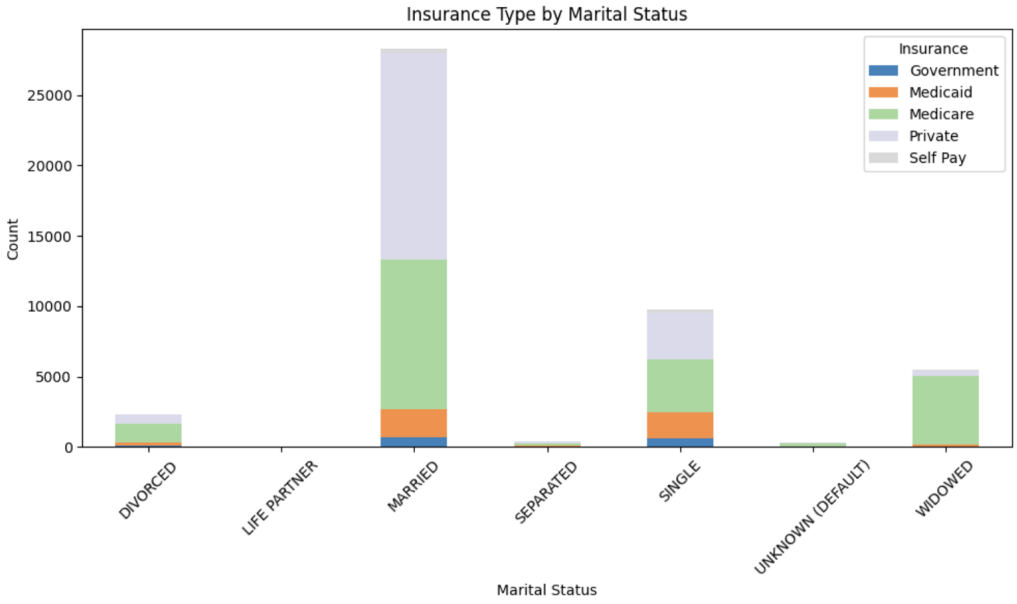
o Multivariate Analysis:



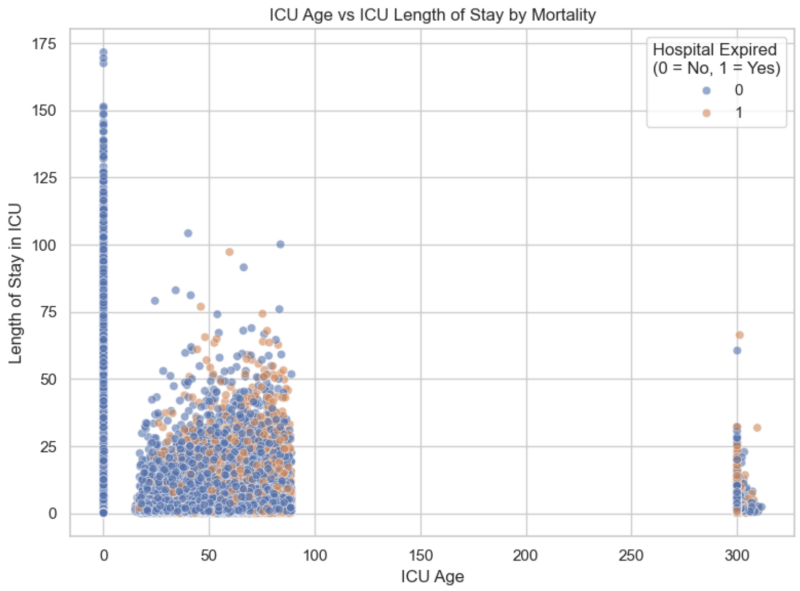
The plot shows that Medicare and Private insurance patients make up the largest share of hospital admissions. Notably, Medicare patients have a visibly higher proportion of hospital deaths compared to other insurance groups. This could reflect the older average age of Medicare recipients and possibly more complex health conditions. In contrast, patients with Private insurance or Medicaid appear to have lower mortality rates.



The chart shows that married patients make up the majority of hospital admissions, which is expected given demographic distribution. However, in terms of relative mortality, categories like widowed and divorced appear to have a higher proportion of deaths compared to their total counts.

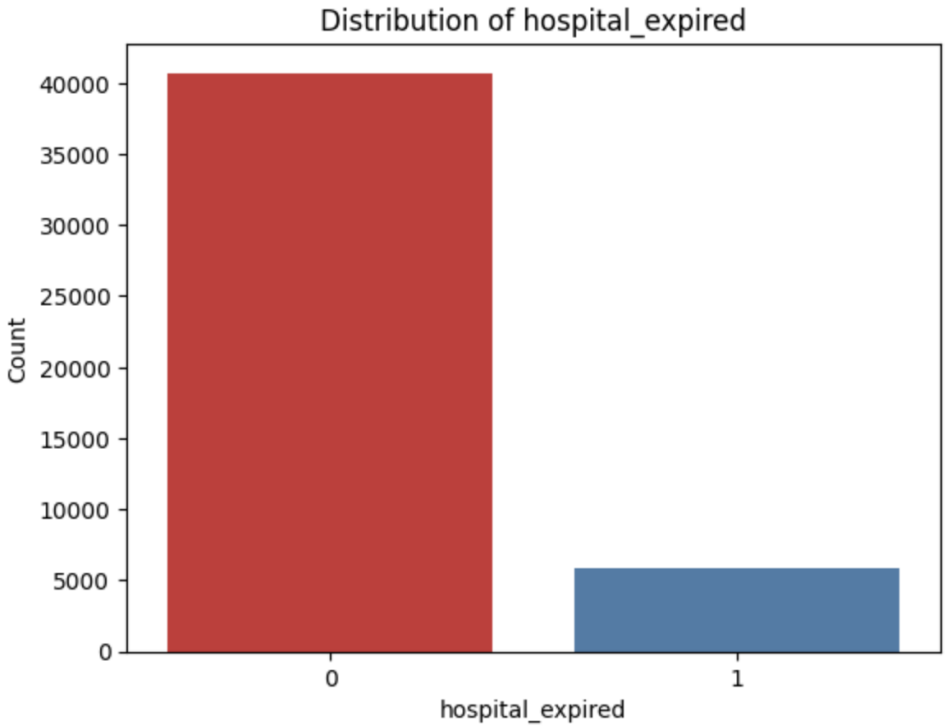


The plot reveals that married and single patients dominate the dataset across all insurance types. Private and Medicare insurance are the most common regardless of marital status, but married individuals are especially likely to have private insurance, while widowed patients rely more heavily on Medicare, likely reflecting their older age.



The scatter plot indicates that most ICU admissions occur between ages 40–90, with a wide range of ICU stay durations.

- **Analysis of Target Variable:**



The target variable hospital_expired is highly imbalanced, with the majority of patients (value 0) surviving their hospital stay, and only a smaller proportion (value 1) marked as deceased. This imbalance is critical to consider in predictive modeling, as it may lead to biased classifiers that favor the majority class. To address this, techniques such as class weighting, or evaluation metrics like AUC-ROC and F1-score should be applied instead of relying solely on accuracy.

Q2. Visualize the patterns of missing data, if any (e.g., using heatmaps or missing value matrices). Discuss potential reasons for the missing data and its potential impact on your analysis and modeling.

	Missing Count	Missing Rate (%)
language	21311	45.81
religion	434	0.93
marital_status	9763	20.99
icu_age	53	0.11
los_icu	53	0.11
mechanical_ventilation	53	0.11
VASOPRESSOR_USE	53	0.11
death_age	40707	87.50

- **death_age** is likely missing for patients who are still alive or were discharged without dying in the hospital.

- **icu_age**, **los_icu, mechanical_ventilation and VASOPRESSOR_USE** may be missing for patients who were never admitted to the ICU during their hospital stay.
- **language, religion** and **marital_status** may be patient-reported, and thus more likely to be incomplete.

Q3. Identify and analyze potential outliers in your numerical features. Briefly outline your initial thoughts on how you might handle missing values and outliers in the subsequent modeling stage (you don't need to implement the handling yet, just discuss potential strategies).

Key numerical features with potential outliers:

- admission_age, icu_age, death_age: Values exceeding 120–130 years are biologically implausible, with some reaching over 300 — likely due to data entry errors. We can cap at biologically realistic maximum (e.g., 110–120 years) or remove values > 150
- los_adm (length of hospital stay): Contains extreme values up to 200+ days, with most patients staying under 15 days. We can apply log transformation or cap at 95th percentile.
- los_icu: Right-skewed, with rare ICU stays beyond 60+ days — possible, but uncommon. We can apply log transformation or cap at 95th percentile.
- admission_count and icu_count: Vast majority have count = 1, but some reach 50+, indicating extreme outliers. We can cap at upper limit (e.g., 10 or 15) or bin into categories (e.g., 1, 2–5, 6+).

Strategy for Handling Missing Values:

- **death_age:** We can create a binary flag–death_age_missing to indicate if it's missing, and use both death_age and death_age_missing as features.
- **Icu_age**: We can create a binary flag–icu_age_missing to indicate if it's missing, and use both icu_age and icu_age_missing as features.
- **language, religion**, **marital_status :** We can use 'UNKNOWN' or 'OTHER' to replace.

1.3 Data Preprocessing

Based on your EDA, perform necessary data preprocessing steps to prepare your data for modeling.

Q4. Describe the data preprocessing steps you have performed so far. Clearly justify your chosen methods. This may include:

o Handling missing values (e.g., imputation, removal)

- **death_age:** We create a binary flag–death_age_missing to indicate if it's missing, and use both death_age and death_age_missing as features.
- **language, religion**, **marital_status :** We use 'UNKNOWN' or 'OTHER' to replace.

o Handling outliers (e.g., capping, transformation, removal).

- **Age features** (adm_age, icu_age): Capped at 100 years to remove unrealistic values.
- **Admission & ICU counts** (admission_count, icu_count): Rows with values > 10 were removed, as these represent extreme rare cases that may skew the model.
- **Length of Stay** (los_adm, los_icu): Negative values were removed. Right-side outliers were capped using the IQR method (above Q3 + 1.5*IQR) to reduce the effect of extreme values.

o Encoding categorical variables (e.g., one-hot encoding, label encoding).

- Applied **one-hot encoding** to features such as: gender, insurance, marital_status, religion, language.

o Feature scaling (e.g., standardization, normalization).

- **Standardization (Z-score scaling)** was applied to continuous numerical features (e.g., adm_age, los_adm, los_icu) for algorithms sensitive to scale (e.g., logistic regression, KNN).

- Not applied to tree-based models (e.g., Random Forest), as they are insensitive to scale.

o Creating new features (feature engineering) if applicable. Explain your reasoning for creating them.

- **language:** Simplified to two categories: 'ENGL' vs 'OTHER'.
- **religion:** Top 5 most frequent values were retained; others grouped into 'OTHER'.
- **ethnicity:** Custom mapping function applied to group into broad categories (WHITE, BLACK, ASIAN, HISPANIC, UNKNOWN, etc.), improving consistency and reducing dimensionality.
- **death_age_missing:** New binary feature created to help model leverage the absence of death_age.

Q5. What is the size and structure of your dataset *after* the preprocessing steps? How does it compare to the original dataset size reported in Milestone 1?

```
📐 Dataset Shape Comparison:
Original shape : (46520, 19)
Processed shape: (36637, 20)

📉 Missing Value Comparison (Top 10 columns with missing values):
                        Original Missing  Processed Missing
death_age                        40707.0              32946
language                         21311.0                  0
marital_status                    9763.0                  0
religion                           434.0                  0
VASOPRESSOR_USE                     53.0                  0
icu_age                             53.0                  0
los_icu                             53.0                  0
mechanical_ventilation              53.0                  0

🧬 Data Type Changes:
                  Original Processed
death_age_missing      NaN     int64

📌 Column Comparison:
Dropped columns : set()
Added columns   : {'death_age_missing'}
```

After preprocessing, the dataset was reduced from 46,520 rows and 19 columns to 36,637 rows and 20 columns, reflecting the removal of records with invalid or extreme values (e.g., implausible ages or negative stays) and the addition of engineered features.

- Missing data was fully addressed: All missing values in features such as language, marital_status, and ICU-related fields were successfully imputed or handled.

- A new feature death_age_missing was created to capture the presence or absence of death_age, helping the model differentiate survivors from the deceased without risking target leakage.

- No columns were dropped, and data types remained mostly stable, indicating consistent formatting.

These preprocessing steps ensure the dataset is now cleaner, more complete, and model-ready.

1.4 Feature Selection and Engineering

To better identify variables associated with in-hospital mortality, we applied both Logistic Regression and Random Forest models, evaluating their performance using Confusion Matrices and AUC-ROC scores.

In this analysis, we excluded the variables death_age and death_age_missing from the dataset to prevent data leakage. Including these variables would compromise the integrity of our models:

- death_age is only known after a patient has died, making it unavailable at the time of prediction. Using it would simulate knowledge of the outcome beforehand.
- death_age_missing serves as a proxy for whether a patient has died (i.e., if death_age exists), which directly leaks information about the target label hospital_expired.

By excluding these variables, we ensure the models rely only on features that would realistically be available before or during hospitalization, preserving their real-world applicability.

As shown in the table below, Random Forest demonstrates overall better predictive ability, with a higher AUC (0.91). It also performs more consistently in terms of precision (0.7) and F1-score (0.58), indicating more reliable predictions. Although it captures slightly fewer death cases (Recall = 0.5), it also results in fewer false positives (155).

Logistic Regression

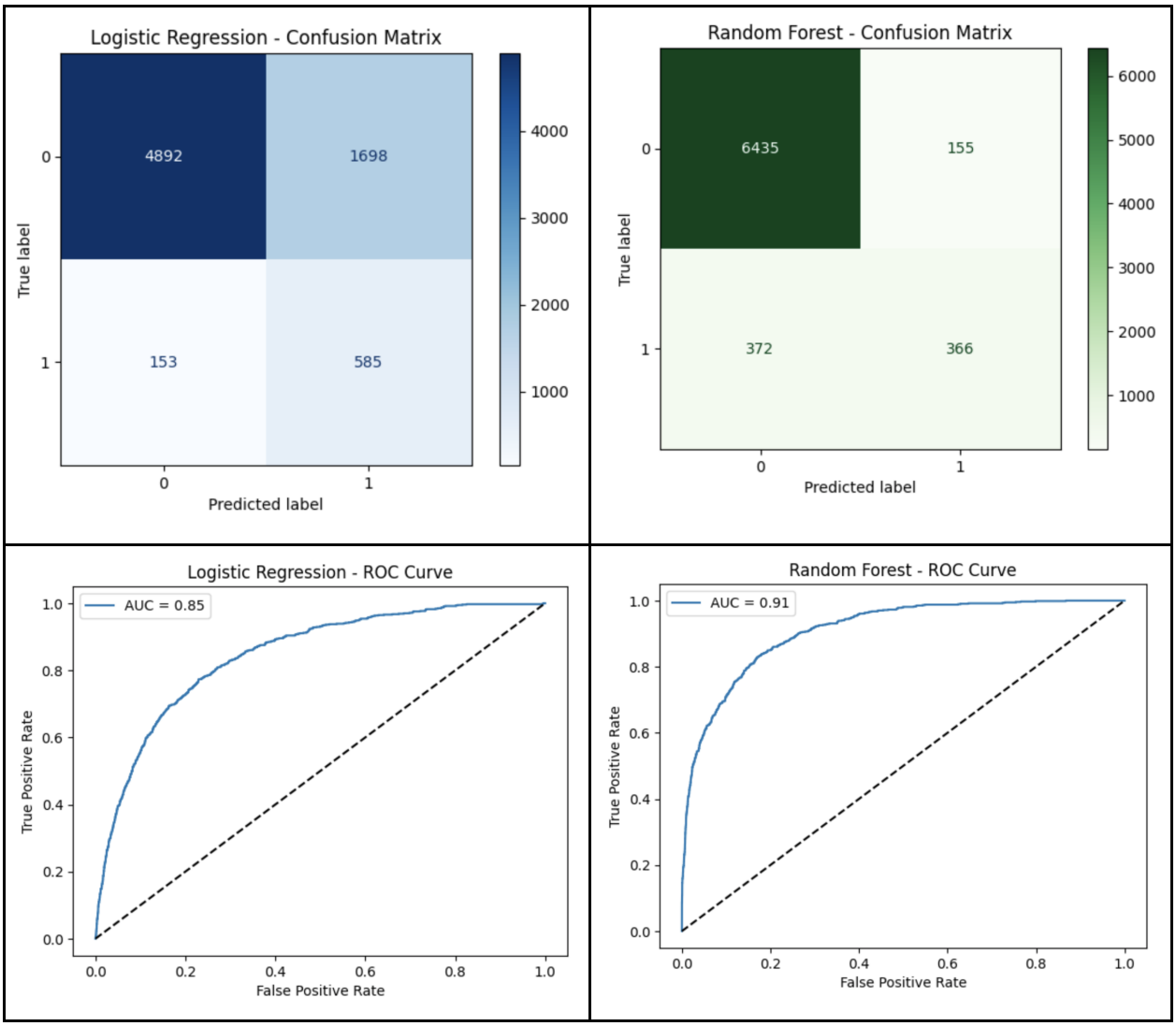
Classification Report:

	precision	recall	f1-score	support
0	0.97	0.74	0.84	6590
1	0.26	0.79	0.39	738
accuracy			0.75	7328
macro avg	0.61	0.77	0.61	7328
weighted avg	0.90	0.75	0.80	7328

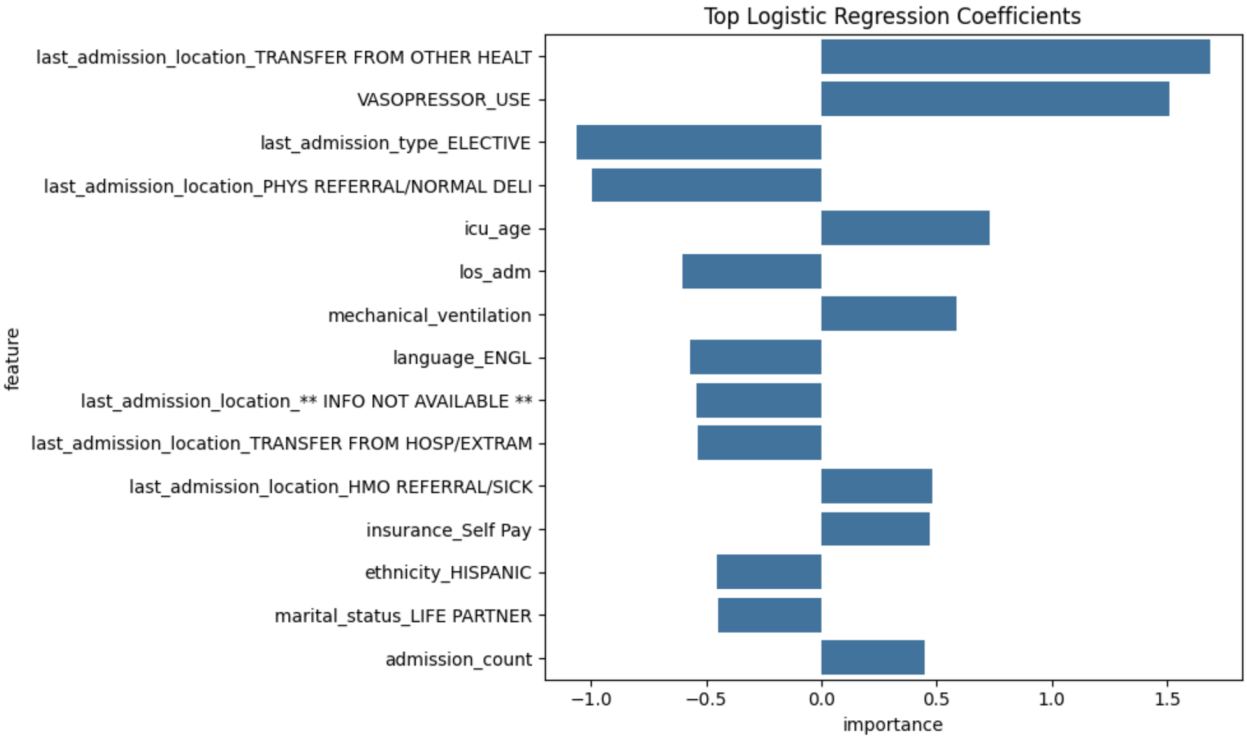
Random Forest

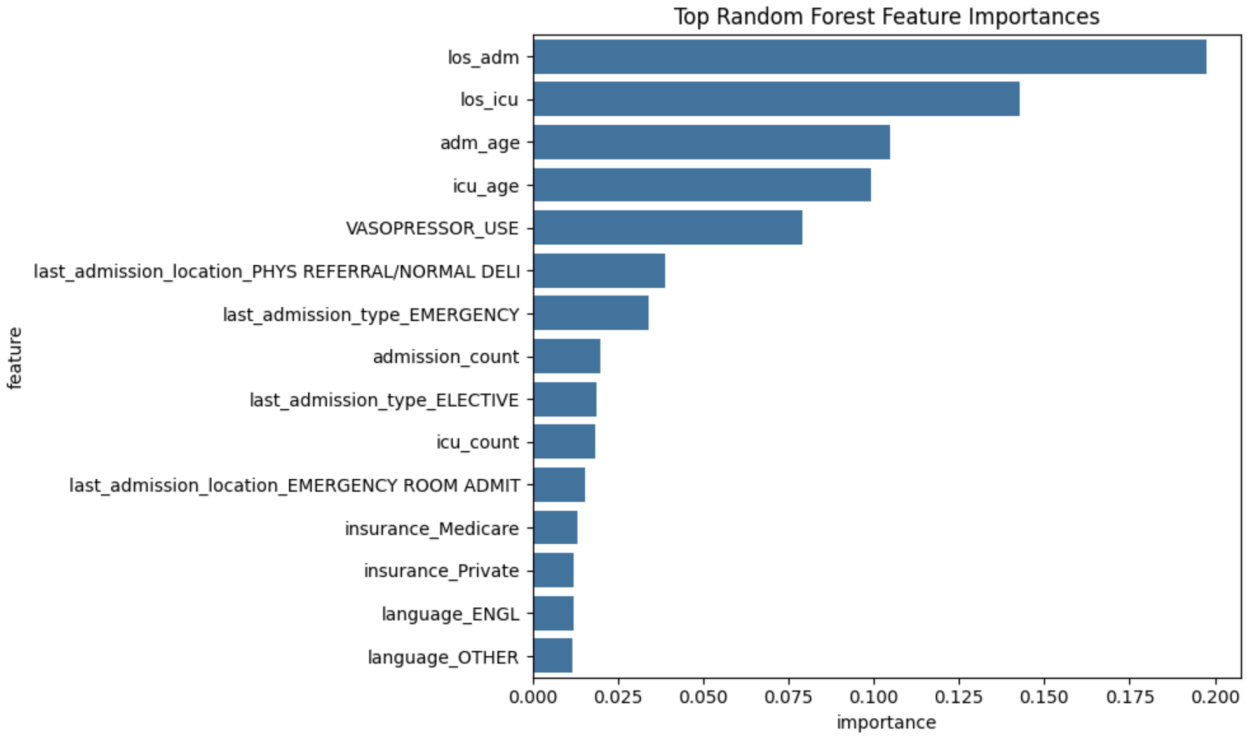
Classification Report:

	precision	recall	f1-score	support
0	0.95	0.98	0.96	6590
1	0.70	0.50	0.58	738
accuracy			0.93	7328
macro avg	0.82	0.74	0.77	7328
weighted avg	0.92	0.93	0.92	7328



Additionally, we found that Logistic Regression and Random Forest both selected seven common features, further demonstrating that the features we identified are reliable for predicting mortality.





Since Random Forest calculates feature importance—typically using Gini Importance—based on how much each feature contributes to splitting across all trees, the total importance values sum up to approximately 1. Most features fall within the 0.01 to 0.1 range. Therefore, any feature with an importance value greater than 0.1 is likely a key factor that the model heavily relies on.

Even if most features have relatively low importance, a well-performing model—as seen in our Random Forest results with high AUC and F1-score—still indicates the selected features are effective. Often, these lower-importance features provide supplementary information that enhances the model's overall performance. As a result, we decided to retain the top 15 most important features from the random forest model for our final model training.

The following are the explanation for the 15 selected features:

1. **los_adm**: the length of stay (LOS) for a patient in a general hospital ward or department after admission.
2. **los_icu**: the length of stay (LOS) in the Intensive Care Unit (ICU) specifically.
3. **adm_age**: the age of the patient at the time of their admission to the hospital.
4. **icu_age**: the age of the patient at the time of their admission to the ICU.
5. **VASOPRESSOR_USE**: whether the patient required the use of vasopressors
6. **last_admission_location_PHYS REFERRAL/NORMAL DELI**: whether the location from which the patient was admitted during their last hospital admission are PHYS REFERRAL/NORMAL DELI. "Phys Referral" refers to a referral from a physician, and "Normal Deli" refers to a standard or routine department.
7. **last_admission_type_EMERGENCY**: whether the type of admission during the patient's last visit to the hospital, specifically indicating an emergency admission.
8. **admission_count**: the total number of times a patient has been admitted to the hospital, indicating their overall hospital usage.
9. **last_admission_type_ELECTIVE**: whether the patient's last hospital admission was elective, meaning the admission was scheduled in advance (as opposed to emergency or urgent admissions).
10. **icu_count**: the number of times a patient has been admitted to the ICU. It helps track how often a patient has needed intensive care.
11. **last_admission_location_EMERGENCY ROOM ADMIT**: whether the location from which the patient was admitted during their last visit to the hospital, specifically the Emergency Room (ER).
12. **insurance_Medicare**: whether the patient is covered under Medicare, a U.S. government health insurance program for individuals aged 65 and older, or for those with certain disabilities.
13. **insurance_Private**: whether the patient has private health insurance, which is typically provided by private insurance companies or employers.
14. **language_ENGL**: whether the patient's primary language is English.
15. **language_OTHER**: whether the patient's primary language is other than English, capturing potential language diversity among patients.

2. Modeling

2.1 Preliminary Modeling (Exploratory)

Q6. Based on your understanding of the problem (from Milestone 1) and your EDA findings, select one model that you think is suitable for the problem. Treat this model as your baseline setup. This could be simple ML models (e.g., linear regression, logistic regression).

Our task is a binary classification problem, as the target variable hospital_expired indicates survival (0) or death (1). Logistic Regression is a strong baseline due to its simplicity, interpretability, and effectiveness with linear relationships. It's widely used in healthcare and handles class imbalance(with most patients surviving) reasonably well when using class weights. Given our numerical features and class imbalance observed in the EDA, where we mentioned features like age, counts, and lengths of stay, Logistic Regression is a suitable starting point.

Q7. Briefly describe the steps you took to implement these initial models (e.g., splitting data into training and testing sets, grid search, fitting the model, making predictions). Note that tour code is reusable.

To implement the initial models, I followed several key steps to prepare the data, train the model, and evaluate its performance:

1. **Data Loading and Feature Selection:** Loaded the dataset with pd.read_csv() and selected features by dropping subject_id and keeping hospital_expired as the target.
2. **Data Splitting:** Split the data into training and testing sets (80-20) using train_test_split() with random_state=42 for reproducibility.
3. **Feature Scaling:** Applied StandardScaler() to standardize features, fitting on the training set and transforming both sets.
4. **Class Imbalance Handling:** Used compute_class_weight() to calculate balanced class weights and passed them to the logistic regression model.
5. **Model Training:** Initialized LogisticRegression() with class weights and trained it on the scaled training data.
6. **Model Hyperparameter Tuning:** Used GridSearchCV to find the best hyperparameter, such as regularization strength.
7. **Model Prediction:** Used the best model to predict on the test set.
8. **Model Evaluation:** Assessed performance using classification report, confusion matrix, and AUC-ROC score.
9. **Cross-Validation:** Performed 5-fold cross-validation to evaluate model consistency using accuracy scores.

Q8. Evaluate the performance of these initial models using appropriate evaluation metrics relevant to your problem type (e.g., accuracy, precision, recall, F1-score

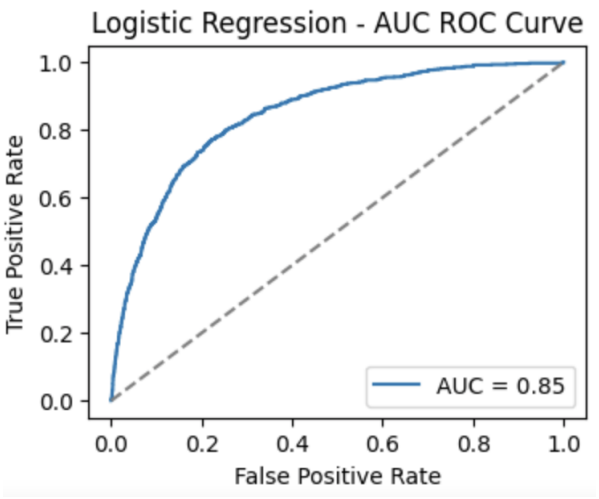
for classification; RMSE, MAE for regression). Report cross-validation performance of evaluation metrics as well as test data performance.

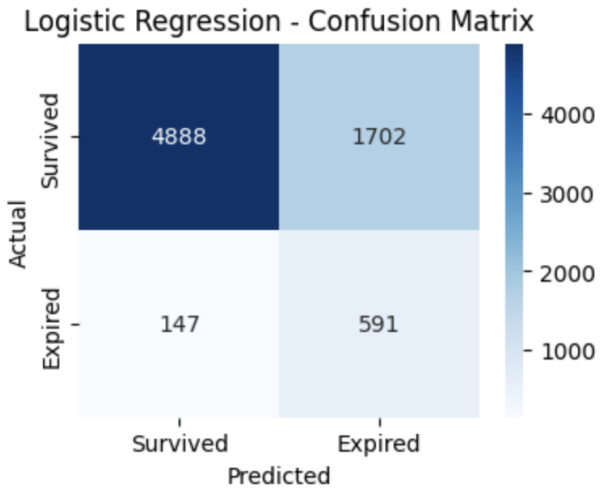
1. Accuracy:
   The model achieved an accuracy of 75% on the test data, indicating that 75% of the predictions were correct. The mean cross-validation accuracy across 5 folds was approximately 75%, suggesting stable and consistent performance.
2. Precision, Recall, F1-score:
   - For class 0 (non-expired patients):
     Precision = 0.97, Recall = 0.74, F1-score = 0.84
   - For class 1 (expired patients):
     Precision = 0.26, Recall = 0.80, F1-score = 0.39
     These results highlight the model's strong ability to recall expired patients (high sensitivity), though with lower precision—indicating a higher rate of false positives.

```
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.74      0.84      6590
           1       0.26      0.80      0.39       738

    accuracy                           0.75      7328
   macro avg       0.61      0.77      0.62      7328
weighted avg       0.90      0.75      0.80      7328
```

```
Cross-Validation accuracy Scores: [0.7517912  0.75759127 0.73984988 0.74769703 0.74270602]
Mean Cross-Validation accuracy: 0.7479270797005413
```

3. AUC-ROC Score:
   The model's AUC-ROC score is 0.85, showing strong discriminative power between the two classes. AUC close to 1 reflects good performance in ranking positive vs. negative classes.
4. Confusion Matrix:
   The model tends to classify more patients as at risk of expiration, resulting in relatively more false positives(1702) but fewer false negatives(147)—prioritizing sensitivity, which may be valuable in medical decision-making.





Overall, the model prioritizes recall for the positive class (expired patients), which may be suitable in a healthcare context where identifying high-risk patients is critical.

However, the low precision for class 1 highlights the need for improvements—potentially through better feature engineering, resampling techniques (e.g., SMOTE), or using more complex models.

Q9. What are your initial observations and insights from these preliminary modeling results? Were the results as expected based on your EDA? What are the potential limitations of the initial model for your problem?

1. Initial Observations & Insights

- Imbalanced Performance Across Classes:
  The model performs much better on class 0 (survivors) than class 1 (expired), consistent with the class imbalance found during EDA.
- Strong Recall for Expired Patients:
  With 80% recall for class 1, the model effectively identifies most high-risk cases—critical in clinical use. Class weighting helped improve recall.
- Low Precision for Class 1:
  Precision remains low (26%), meaning many predicted deaths are false positives, which could strain clinical resources.
- AUC-ROC Score (0.85):
  A strong score indicates solid overall discrimination ability between survived and expired patients.

These results align with the EDA: class imbalance lowered precision, but predictive patterns were still captured. Key mortality-related features appear influential, as shown by decent recall and AUC-ROC.

2. Potential Limitations of the Initial Model

- Class Imbalance Not Fully Resolved:
  Precision for the minority class remains weak despite class weighting.
- Model Simplicity:
  Logistic Regression may miss non-linear patterns. More complex models (e.g., Random Forest) could improve performance.
- Minimal Hyperparameter Tuning:
  The model used limited tuning(regularization strength). Broader optimization could enhance results.

Q10. Based on your insights from Q9, select a couple of model families that you think could address the limitations you reported. Do a comparative evaluation across the models. Choose the best model, report it, and use it for prediction for solving the problem you began with.

Given the limitations of the logistic regression model—especially its inability to capture non-linear relationships and its lower precision for the minority class—we selected:

- Random Forest (tree-based ensemble, good for feature interactions and robustness)
- XGBoost (known for handling class imbalance effectively and providing robust performance through boosting)

These models are well-suited to:

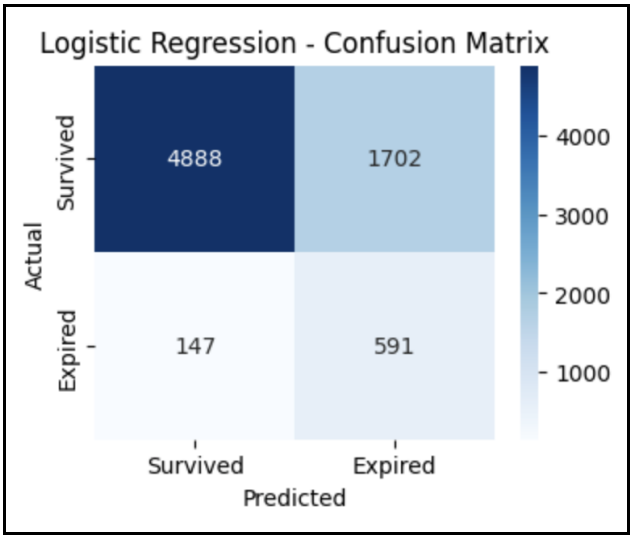
- Improve classification of the minority class (expired patients)
- Capture more complex decision boundaries than logistic regression
- Potentially increase precision and F1-score while maintaining strong recall

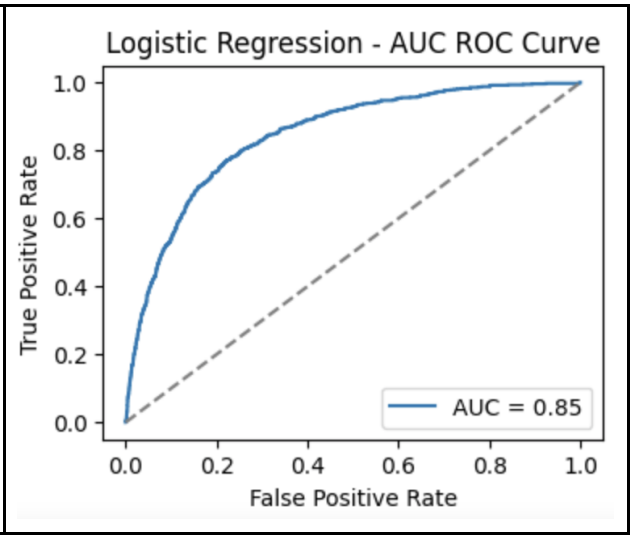
Performance evaluation across Logistic Regression, Random Forest and XGBoost:

Logistic Regression

```
Classification Report:
                precision    recall  f1-score   support

           0       0.97      0.74      0.84      6590
           1       0.26      0.80      0.39       738

    accuracy                           0.75      7328
   macro avg       0.61      0.77      0.62      7328
weighted avg       0.90      0.75      0.80      7328
```

```
Cross-Validation accuracy Scores: [0.7517912  0.75759127 0.73984988 0.74769703 0.74270602]
Mean Cross-Validation accuracy: 0.7479270797005413
```





Classification Report:

- Class 0 (Survived): Precision = 0.97, Recall = 0.74, F1-score = 0.84
- Class 1 (Expired): Precision = 0.26, Recall = 0.80, F1-score = 0.39
- Accuracy: 75%
- AUC-ROC Score: 0.85
- Cross-Validation Mean Accuracy: 74.79%

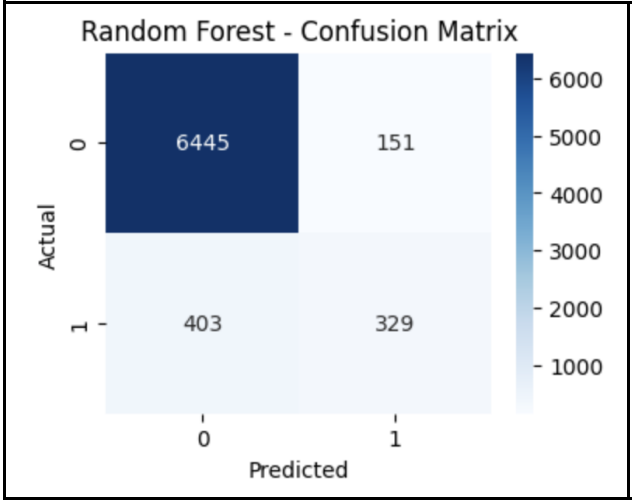
Key Observations:

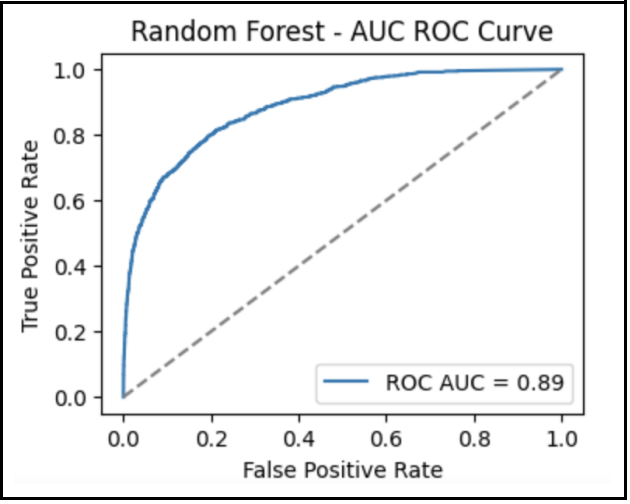
- The model is highly precise for predicting survivors but struggles with precision for predicting expired patients (low precision for class 1).
- It has a strong recall for expired patients (80%), which is valuable for identifying at-risk patients.
- The overall AUC-ROC score is good (0.85), showing decent discriminatory ability between classes.
- The model's performance is fairly consistent across cross-validation folds, with slight fluctuations in accuracy (around 75%).

Random Forest

```
Random Forest - Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.98      0.96      6596
           1       0.69      0.45      0.54       732

    accuracy                           0.92      7328
   macro avg       0.81      0.71      0.75      7328
weighted avg       0.92      0.92      0.92      7328
```

```
Cross-Validation Accuracy Scores: [0.92852269 0.92238144 0.92562265 0.930058   0.92646306]
Mean Cross-Validation Accuracy: 0.9266095688522998
```





Classification Report:

- Class 0 (Survived): Precision = 0.94, Recall = 0.98, F1-score = 0.96
- Class 1 (Expired): Precision = 0.69, Recall = 0.45, F1-score = 0.54
- Accuracy: 92%
- AUC-ROC Score: 0.89
- Cross-Validation Mean Accuracy: 92.66%

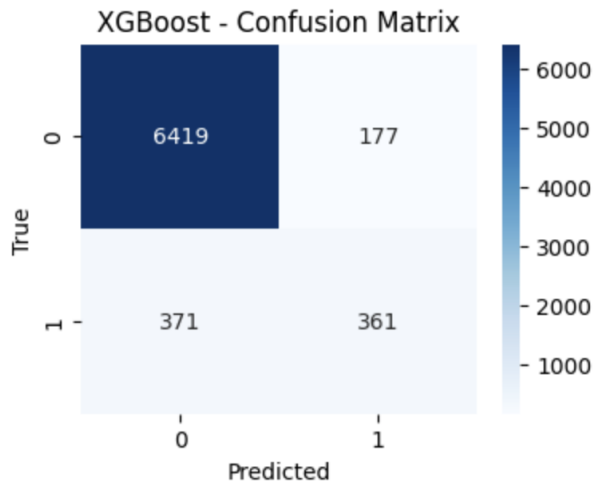
Key Observations:

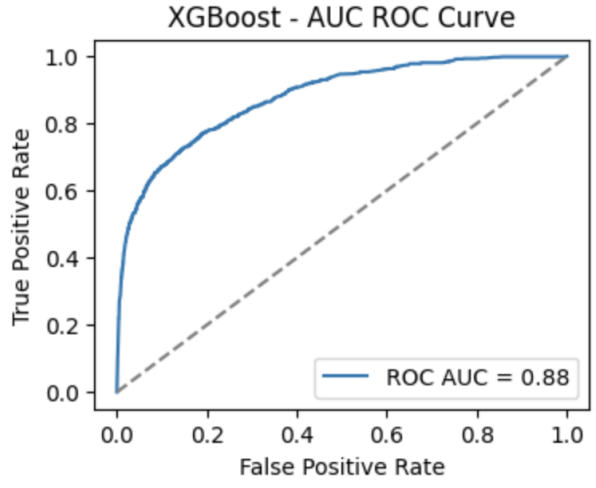
- Random Forest is highly effective at predicting survivors (very high precision and recall for class 0).
- The model struggles more with expired patients, achieving relatively 45% recall and 69% precision.
- The model shows a very strong AUC-ROC score (0.89), indicating better overall discriminatory power compared to logistic regression.
- Cross-validation shows consistent high performance with an average accuracy of about 93%, indicating that the model generalizes well.

XGBoost

```
Classification Report:
                precision    recall  f1-score   support

           0       0.95      0.97      0.96      6596
           1       0.67      0.49      0.57       732

    accuracy                           0.93      7328
   macro avg       0.81      0.73      0.76      7328
weighted avg       0.92      0.93      0.92      7328
```

```
Cross-Validation accuracy Scores: [0.92903446 0.92698738 0.92766974 0.92818151 0.92356253]
Mean Cross-Validation accuracy: 0.9270871225701806
```





Classification Report:

- Class 0 (Survived): Precision = 0.95, Recall = 0.97, F1-score = 0.96
- Class 1 (Expired): Precision = 0.67, Recall = 0.49, F1-score = 0.57
- Accuracy: 93%
- AUC-ROC Score: 0.88
- Cross-Validation Mean Accuracy: 92.71%

Key Observations:

- XGBoost performs similarly to Random Forest, with very high precision and recall for predicting survivors.
- For expired patients, recall is 49%, FI is 57%, which shows better performance than Random Forest in recall and F1.
- The AUC-ROC score is strong but slightly lower than Random Forest (0.88), indicating a bit less discriminative ability.

- Cross-validation results show consistent performance with a mean accuracy of around 93%.

Conclusion:

- **Best Model: XGBoost**

Among the three models, XGBoost emerges as the best choice for mortality prediction due to its balanced performance. While Logistic Regression achieves the highest recall for expired patients (80%), it suffers from very low precision (26%), leading to many false alarms. Random Forest shows excellent overall accuracy (92%) and AUC-ROC (0.89), but its recall for expired patients is the lowest (45%), risking missed critical cases. XGBoost strikes a better balance with a recall of 49% and a higher F1-score (57%) for expired patients, while maintaining strong accuracy (93%) and AUC-ROC (0.88). This trade-off makes XGBoost the most reliable for identifying at-risk patients while minimizing both false positives and false negatives.

Predictive Modeling Using XGBoost: A Machine Learning Pipeline for Mortality Prediction

1. Objective

The goal of this modeling pipeline is to build and evaluate a predictive model to estimate in-hospital mortality (binary classification) using a dataset comprising the top 15 important features.

2. Data Preparation

- **Dataset Loading:** The dataset top15_rf_features_dataset.csv is loaded into a pandas DataFrame. The target variable is hospital_expired, while subject_id is dropped as it is an identifier and not relevant for prediction.
- **Feature-Target Split:** Independent variables (X) are separated from the dependent binary target (y).

3. Data Preprocessing

- **Train-Test Split:** The dataset is split into training and test sets with an 80/20 ratio using a fixed random_state for reproducibility.
- **Feature Scaling:** StandardScaler is applied to normalize the features. This is particularly important for algorithms like XGBoost that can benefit from scaled input for better convergence during optimization.

4. Class Imbalance Handling

- **Class Weights Calculation:** `compute_class_weight` is used to calculate balanced weights, and the ratio for the positive class (`scale_pos_weight`) is passed to the XGBoost classifier.

5. Model Training and Hyperparameter Tuning

- **Model Definition:** An `XGBClassifier` is defined with the calculated `scale_pos_weight` and relevant evaluation metric (`log loss`) for binary classification.
- **Grid Search:** A grid search is performed over a range of hyperparameters (learning rate, maximum tree depth, number of estimators) using 5-fold cross-validation to identify the optimal model configuration.
- **Best Model:** The best hyperparameters identified by `GridSearchCV` are printed and used for evaluation.

6. Model Evaluation

- **Prediction:** The final model makes predictions on the test set, both as class probabilities and binary class outputs.
- **Metrics:**
  - **Classification Report:** Provides precision, recall, f1-score, and support for each class.
  - **Confusion Matrix:** Summarizes true positives, false positives, true negatives, and false negatives.
  - **AUC-ROC Score:** Evaluates the model's ability to distinguish between the two classes.
  - **Cross-Validation Accuracy:** Offers insights into model performance stability and generalization through additional 5-fold CV on the training data.

7. Visualization

- **Confusion Matrix Heatmap:** A heatmap is plotted using seaborn to visualize the confusion matrix for quick interpretation of prediction accuracy.
- **ROC Curve:** The ROC curve is plotted to illustrate the trade-off between sensitivity and specificity, with the AUC score annotated.

Conclusion:

This pipeline effectively integrates data preprocessing, class imbalance handling, model training with hyperparameter tuning, and comprehensive evaluation. XGBoost, combined with GridSearchCV and class weighting, serves as a robust classifier for

predicting hospital mortality. The results provide a reliable foundation for clinical decision support systems and further model refinement.