# Applying ML and DL to MIMIC Data for Mortality Prediction

Sharang Agarwal (sa62567)

# Introduction & Objectives

**Assignment Goal:** Familiarize with ML/DL techniques for analyzing EHR data and deriving insights for healthcare risk management.

**Project Task:** Predict in-hospital mortality using MIMIC III data.

**Learning Outcomes:**

- Apply ML/DL to EHR data.
- Apply ML/DL to identify basic problems in healthcare (like mortality prediction).

# Data Sources

**Dataset:** MIMIC III Clinical Database

**Tables Used:**

- PATIENTS.csv.gz: Patient demographic information (DOB, GENDER).
- ADMISSIONS.csv.gz: Admission details (ADMITTIME, DISCHTIME, admission type, insurance, etc.), hospital expiration flag.
- DIAGNOSES_ICD.csv.gz: ICD-9 diagnosis codes for each admission.
- ICUSTAYS.csv.gz: ICU stay information (care unit).

# Feature Engineering (part 1) - Core Patient & Admission Info

**Target Variable (Mortality):** Defined using HOSPITAL_EXPIRE_FLAG from the ADMISSIONS table.

**Length of Stay (LOS):** Calculated from DISCHTIME and ADMITTIME (converted to days). *Note: While calculated, LOS is used as a feature for mortality prediction in this code.*

**Age:** Calculated using patient DOB and ADMITTIME.

**Patient Demographics:** Merged GENDER from the PATIENTS table.

**Diagnoses (ICD-9 Codes)**

- **Processing:**
  - Filtered non-alpha codes.
  - Extracted the first 3 digits.
  - Mapped codes to broader categories (e.g., 'infectious', 'neoplasms', 'circulatory') using predefined ranges.
- **Aggregation:** Grouped diagnosis categories by hospital admission (HADM_ID).
- **Transformation:** Created binary indicator variables (dummy variables) for each diagnosis category per admission.

# Feature Engineering (part 2) - ICU Stays & Merging

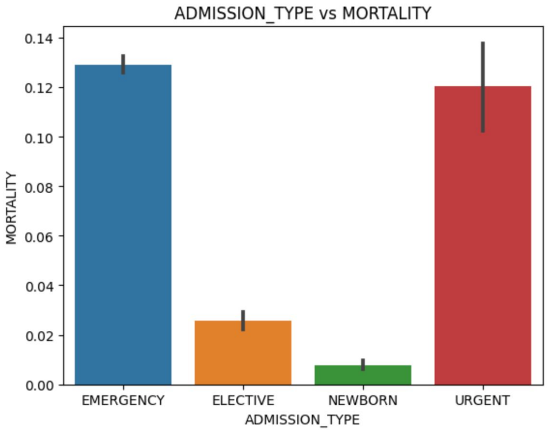
**ICU Care Unit:** Simplified FIRST_CAREUNIT into 'ICU' category.

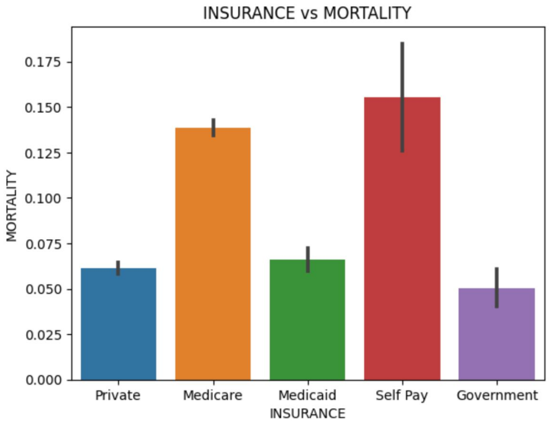
- **Aggregation:** Grouped ICU stays by hospital admission (HADM_ID).
- **Transformation:** Created binary indicator for ICU stay per admission.
- **Data Merging:** Merged processed features from Admissions, Patients, Diagnoses, and ICU stays into a single dataframe based on HADM_ID.

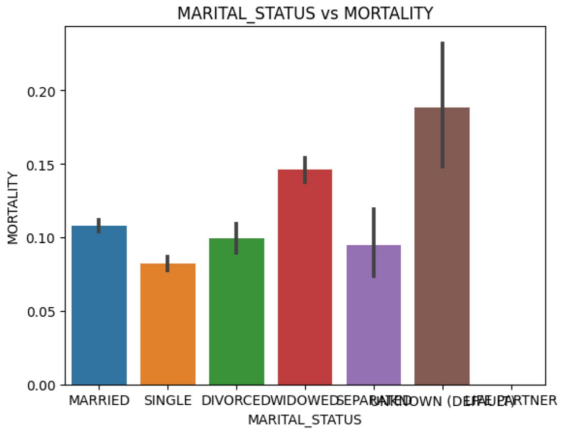
**Create one-hot encoding:**

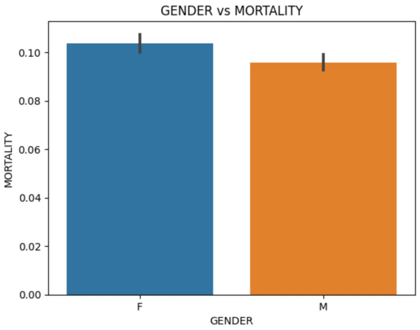
- **Categorical Variables:** Converted features into numerical representations using one-hot encoding.
  - ADMISSION_TYPE
  - INSURANCE
  - RELIGION
  - MARITAL_STATUS
  - ETHNICITY
  - GENDER

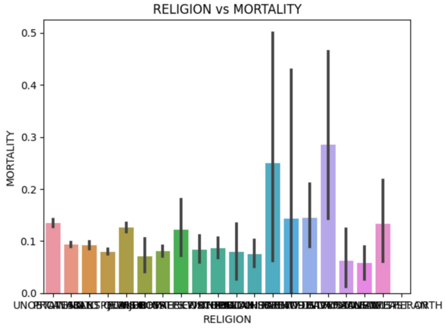
# Exploratory Data Analysis - Categorical features

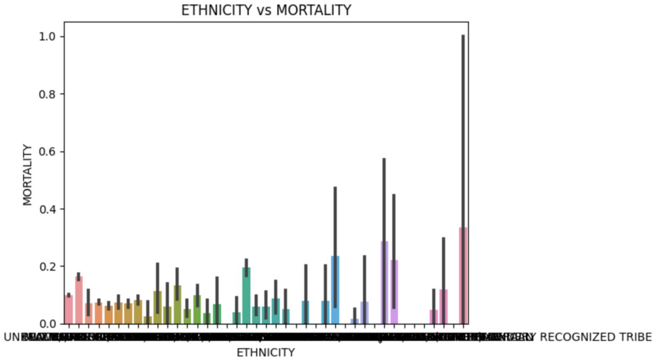


# Feature Engineering (part 2) - ICU Stays and One-hot encoding

**ICU Care Unit:** Simplified FIRST_CAREUNIT into 'ICU' category.

- **Aggregation:** Grouped ICU stays by hospital admission (HADM_ID).
- **Transformation:** Created binary indicator for ICU stay per admission.
- **Data Merging:** Merged processed features from Admissions, Patients, Diagnoses, and ICU stays into a single dataframe based on HADM_ID.

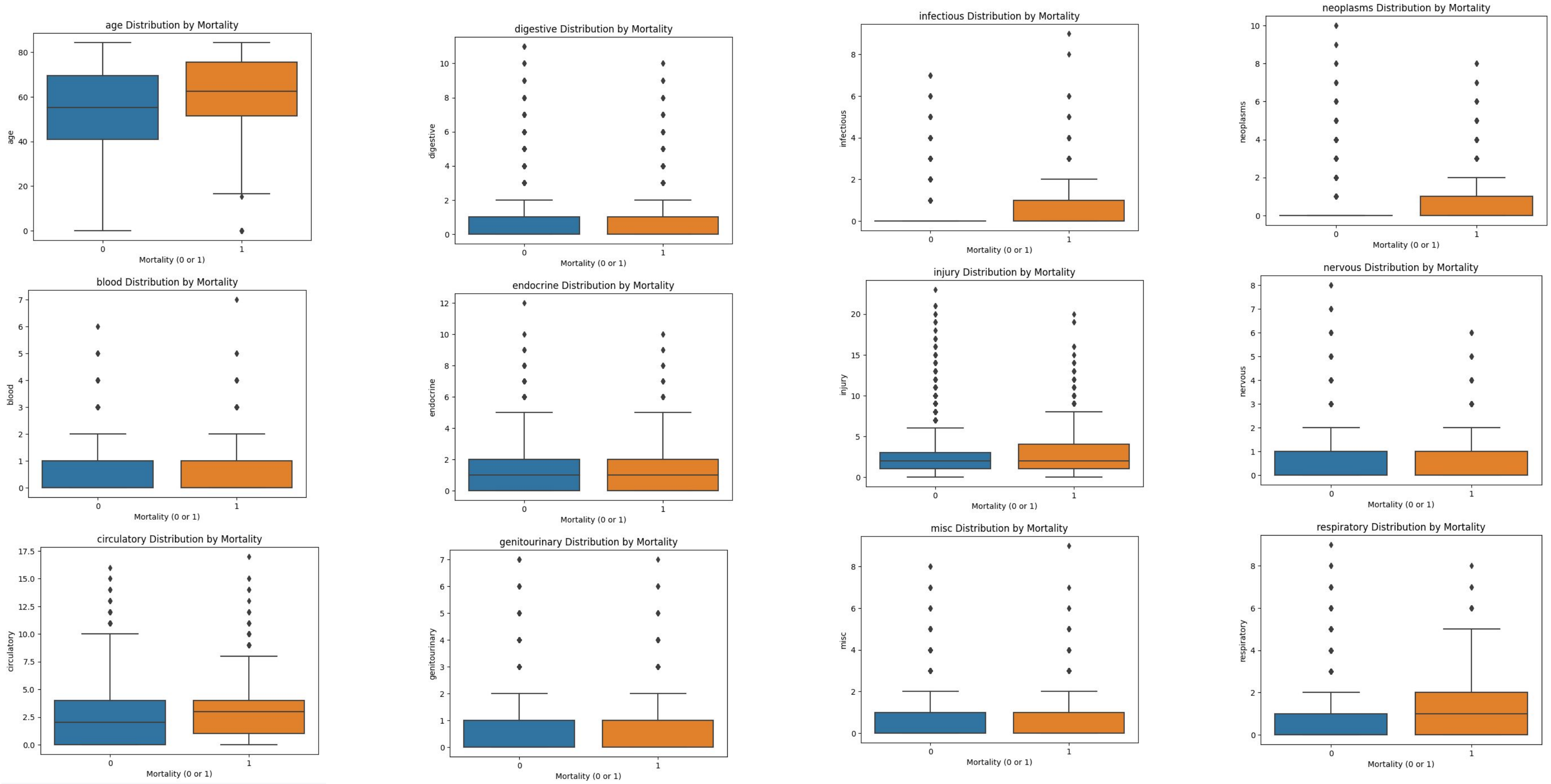
**Create one-hot encoding:**

- **Categorical Variables:** Converted features into numerical representations using one-hot encoding.
  - ADMISSION_TYPE
  - INSURANCE
  - RELIGION
  - MARITAL_STATUS
  - ETHNICITY
  - GENDER

# Data Cleaning & Preprocessing - Feature Selection & Scaling

- **Feature Selection:**
  - Calculated correlations between numerical features and the 'MORTALITY' target.
  - Selected features with a correlation magnitude greater than 0.005.
  - Defined final feature_list.
- **Outlier Handling (Age):** Capped age at the 90th percentile, replacing higher values with the mean.
- **Handling Missing Values:** Dropped rows with missing values in the selected feature_list .
- **Feature Scaling:** Applied MinMaxScaler to scale all features in the final list to a range between 0 and 1.

# Exploratory Data Analysis - Final Feature list vs. Mortality (Few diagrams below)



# Model training to predict mortality

**Data Splitting**

- **Method:** Split the data into training (80%) and testing (20%) sets.
- **Variables:**
  - X_train, X_test: Scaled features for training and testing.
  - y_train_mortality, y_test_mortality: Target variable (Mortality) for training and testing.

**Model 1 - Gradient Boosting Regressor**

- **Purpose:** Predict mortality (treated as regression target for GBR in the code, although evaluation uses classification metrics).
- **Implementation:** Used sklearn.ensemble.GradientBoostingRegressor.
- **Training:** Fit the model on the training data (X_train, y_train_mortality).

**Model 2 - Deep Learning (Neural Network)**

- **Architecture:**
  - Input Layer: Shape corresponding to the number of features.
  - Hidden Layers: Dense(128, relu), Dense(64, relu), Dense(32, relu).
  - Output Layer: Dense(1, sigmoid) for binary classification.
- **Compilation:**
  - Optimizer: Adam (learning_rate=0.01).
  - Loss: Binary Crossentropy.
  - Metrics: Accuracy.
- **Training:** Trained for 10 epochs with a batch size of 32.

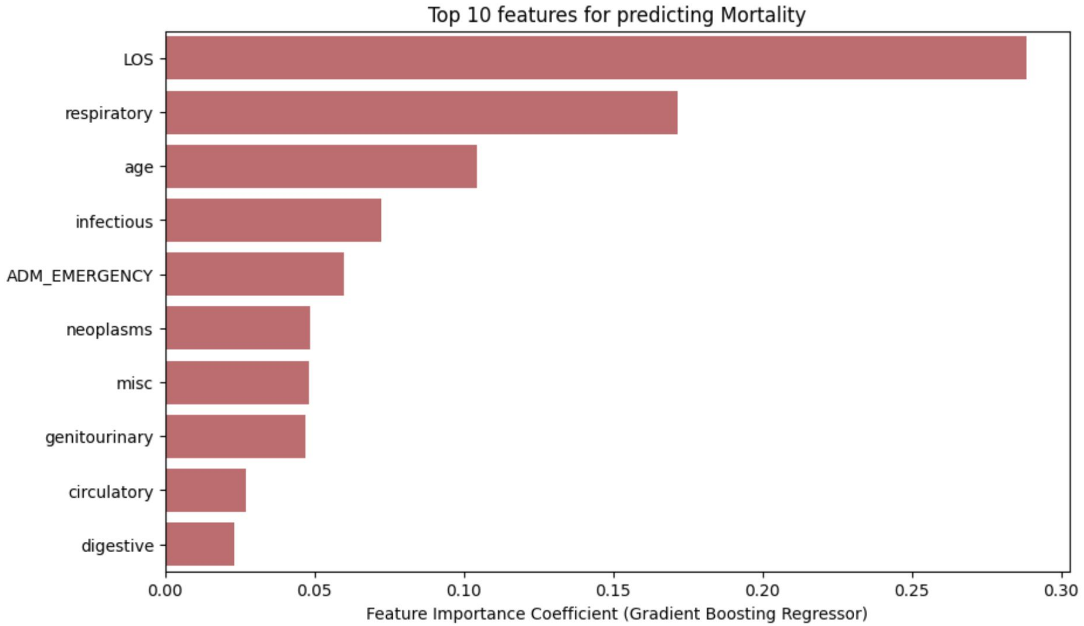
# Model Evaluation Results for Mortality Prediction

GBM performed slightly better in the iterations.

- **GBM:**
  - AUC-ROC: 0.8496
  - Mean Squared Error: 0.0713
  - R-squared: 0.2182

- **Neural Network:**
  - AUC-ROC: 0.8472
  - Mean Squared Error: 0.0736
  - R-squared: 0.2182

Extracted feature importances from GBM Model



# Code Base link

- https://github.com/sharangagarwal/msai_ai_healthcare/blob/main/assignment_MIMIC_ML_DL.ipynb