Report on Assignment 2: Predicting Length of Stay in ICU Patients with Digestive System Diseases

Fabian Alexander Bug (ID: 202412325)
Gabriele Pascali (ID: 202411175)
Giulio Tamburini (ID: 202411174)

Machine Learning Course 2025
Supervisors: Ines Dutra, Maria Pedroto
University of Porto
June 8th, 2025

Contents

1. Introduction

1.1. Objective of the Assignment

The goal of this assignment is to develop a predictive model for estimating the Length of Stay (LOS) in the Intensive Care Unit (ICU) for patients diagnosed with diseases of the digestive system. Accurate early prediction of LOS can support hospitals in resource allocation, scheduling, and treatment planning.

.

1.2. Importance of LOS Prediction

ICU resources are both limited and subject to cost. Early LOS prediction supports clinical decision-making by identifying patients who are likely to require longer stays. This enables efficient resource planning.

2. Dataset Overview

2.1. MIMIC-III Description

We use the MIMIC-III database, a publicly available ICU dataset that includes over 40,000 anonymized patient records. We filtered this dataset to select adult patients diagnosed with diseases of the digestive system, according to the ICD-9 classification (codes 520–579).

2.2. Relevant Tables

We primarily used the following tables:

- PATIENTS.csv:
  Contains basic information about each patient: SUBJECT_ID, gender (GENDER), date of birth (DOB), date of discharge or death (DOD). This table defines the patient and serves as the primary reference. MIMIC includes nearly 38,600 adult patients aged 16 and older.

- ADMISSIONS.csv:
  Information on hospital stays, including HADM_ID (Hospital Admission ID), admission date (ADMITTIME), discharge date (DISCHTIME), in-hospital mortality (HOSPITAL_EXPIRE_FLAG), and reason for admission (ADMISSION_TYPE). In addition to patient data, department and insurance information is also provided.

- ICUSTAYS.csv:
  Details on intensive care unit stays, linked via ICUSTAY_ID, HADM_ID, and SUBJECT_ID. Provides data such as admission times (INTIME), length of stay (LOS), and the respective care unit (FIRST_CAREUNIT, LAST_CAREUNIT). Basis for filtering ICU stays from the Metavision system.

- DIAGNOSES_ICD.csv:
  Contains ICD-9 diagnoses linked to HADM_ID for each hospital stay. Each entry

contains ICD9_CODE and SEQ_NUM, which specifies the order of diagnoses, with SEQ_NUM = 1 typically indicating the primary diagnosis. This table was used to filter for ICD-9 codes in the range 520–579, corresponding to diseases of the digestive system.

- INPUTEVENTS_MV.csv:
  Records administered fluids and medications via the Metavision system. Data points include START TIME, ITEM ID, AMOUNT, and AMOUNT UOM. They are precisely documented in terms of time and form the basis for analyzing the first 12 hours of ICU stays.

- D_ICD_DIAGNOSES.csv:
  Displays the ICD-9 diagnoses: ICD9_CODE and LONG_TITLE (description). This “dictionary table” enables semantic interpretation of the diagnoses in the dataset.

- D_ITEMS.csv:
  A reference work for ITEMIDs in chart or input/output tables. Contains ITEMID, LABEL, DBSOURCE, and much more. By mapping ITEMID to LABEL, it is possible to see, for example, which medication was administered in relation to a specific record.

3. Data Preprocessing

In the data preprocessing phase, we performed several steps to prepare the MIMIC-III dataset to get the best results possible. Our goal was to create a high-quality, clean, and consistent dataset that serves as good foundation for our predictive models.

3.1. Filtering and Diagnosis Extraction

Initially, we filtered the dataset by selecting patients with a specific diagnosis code. However, to improve model generalizability and capture a broader clinical spectrum, we later switched to selecting a group of related ICD-9 codes instead of a single code. In particular, we used the group `520--579`, corresponding to diseases of the digestive system, as defined in the official ICD-9 classification. This approach allowed the model to learn patterns across a more diverse set of patients within the same general disease category. The selected patients were then merged with the `ADMISSIONS.csv`, `ICUSTAYS.csv`, and `PATIENTS.csv` tables to incorporate information about hospital admissions, ICU stays, and patient demographics.

3.2. Handling Missing Values and Outliers

An essential step in preprocessing involved managing missing values and outliers:

- **Anonymized Birthdates:** To ensure data consistency and avoid bias, we excluded patients with anonymized birthdates specifically, those aged over 89 at any point in the database. According to HIPAA regulations, these patients are considered potentially identifiable and therefore have their date of birth artificially shifted 300 years back from their first hospital admission. Since this data manipulation could introduce inconsistencies, we removed all such patients from the analysis.

- **Age Restriction:** We limited our analysis to adult patients (age $\geq$ 18 years), as treatment methods and medical protocols potentially differ between adults and minors
- **LOS Outlier Removal:** Patients with extremely short ICU stays (less than 0.5 days or 12 hours) or exceptionally long stays (greater than 70 days) were also excluded. We consider them as outliers and will not consider them to train our model.

3.3. Age Grouping and Demographics

To give us the opportunity to better interpret our findings and identify potential patterns across different age groups, we created patient groups into the following age brackets:

- 18–29 years
- 30–49 years
- 50–69 years
- 70–89 years

This grouping enabled analysis of LOS and other demographic characteristics, such as mortality rates and average medication usage.

3.4. Age-Based Patient Distribution and LOS

3.4.1 Patient Count by Age Group

Age Group	**Number of Patients**
18–29	522
30–49	3487
50–69	10459
70–89	8188

Table 1: Patient count grouped by age category

3.4.2 Average LOS by Age Group

Age Group	**Average LOS (days)**
18–29	5.27
30–49	4.45
50–69	4.75
70–89	4.28

Table 2: Average Length of Stay (LOS) grouped by age category

3.4.3 Interpretation

The distribution of patients across age groups (Table 1) shows that the majority of ICU admissions for patients with infectious and parasitic diseases fall within the older age brackets:

- More than 4/5 of patients are aged 50–89, indicating that ICU admissions for these conditions are predominantly a concern in older adults.
- The 18–29 age group represents only a small fraction of the dataset.

Interestingly, when analyzing the average LOS by age group (Table 2), a different trend emerges:

- The 50–69 age group has the **highest average LOS** (4.75 days), followed closely by 30–49 (4.45 days).
- The oldest group (70–89) actually has the **shortest average LOS** (4.28 days) among all age brackets above 30.

This counterintuitive result may suggest that younger patients admitted to the ICU with infectious or parasitic diseases may experience more severe or prolonged illness episodes requiring extended treatment.

These observations show that incorporating demographic variables such as age might be useful. However, further analysis is needed to assess whether they significantly contribute to the observed LOS patterns.

3.5. ICU Mortality Considerations

Figure 1 presents a comparison of the number of ICU survivors and deceased patients across different age groups. As the chart indicates, the majority of patients in each group survive their ICU stay, with survival counts consistently exceeding mortality counts. This pattern holds across all age categories, including the oldest group (70–89), which shows the highest absolute number of deceased patients.

The number of deaths increase with age, particularly in the 50–69 and 70–89 groups, reflecting the expected correlation between age and ICU mortality. In contrast, mortality in the 18–29 group is very low.

To maintain a better predictability, we decided to drop deceased patients in our dataset.

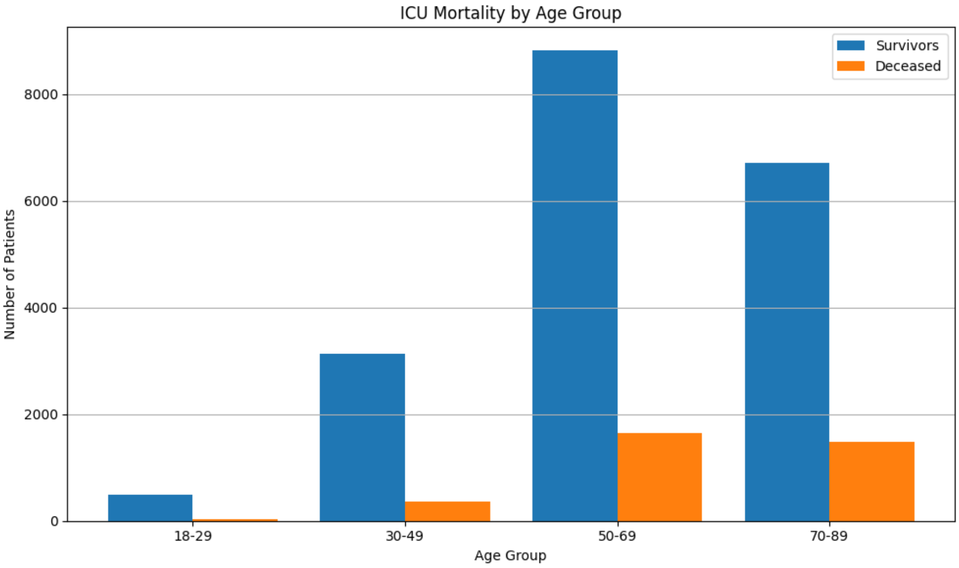

Figure 1: ICU Mortality by Age Group: Comparison of number of survivors and deceased patients.

3.6. Medication Processing

To incorporate clinical intervention data into our model, we processed medication administration records from the MetaVision system, using `INPUTEVENTS_MV.csv`. First, we removed administrative columns such as comments and timestamps unrelated to actual medication delivery. Then, we merged these input events with ICU metadata (`ICUSTAY_ID`, `HADM_ID`, `SUBJECT_ID`, and `INTIME`) to determine the time elapsed since ICU admission for each entry. We left only those events that occurred within the first 12 hours of ICU stay, aligning with our goal of early prediction.

We used the `D_ITEMS.csv` table to enrich the dataset with descriptive labels for each `ITEMID`. We filtered the items to those recorded in MetaVision and with non-null labels.

A key challenge was that many `ITEMID`s appeared with inconsistent units of measurement (e.g., grams, milligrams, micrograms, liters, milliliters). To standardize the `AMOUNT` values, we defined explicit unit conversion rules:

- grams to milligrams
- micrograms to milligrams
- liters to milliliters

After applying these transformations, we identified `ITEMID`s that still appeared with multiple units and retained only those rows associated with the dominant (most frequent) unit for each `ITEMID`. This eliminated inconsistencies and ensured uniformity in measurement.

For each ICU stay, we aggregated the total amount administered per `ITEMID` and pivoted the data so that each item became its own feature (e.g., `ITEM_220045`). Additionally, we calculated two summary features:

- `PATIENTWEIGHT`: first recorded weight during ICU stay

- `NUM_UNIQUE_ITEMS`: number of distinct medications administered within the first 24 hours

Missing values in drug quantities were imputed with 0 (no administration), while missing patient weight was filled using the median. This cleaned and structured medication data was merged into the main DataFrame, forming an essential part of our feature set for modeling.

3.7. Feature Engineering

The quality of predictive modeling significantly depends on the quality of input features. To summarize the new features regarding medical data:

- **Prior ICU Stays:** We calculated the number of previous ICU stays per patient (feature `NUM_PRIOR_ICU_STAYS`) to better represent patient health status and medical history.
- **Medication Features:** From `INPUTEVENTS_MV.csv`, we extracted medication and fluid administration data from the first 12 hours after ICU admission. To ensure data consistency, we standardized units—for example, converting grams to milligrams and liters to milliliters.
- **Medication Aggregation:** We summed the total administered amounts of medications per `ITEMID` and counted the number of different medications administered to each patient during the first 12 hours (feature `NUM_UNIQUE_ITEMS`).
- **Patient Weight:** Each patient's weight was extracted from the earliest ICU records, and missing values were replaced by the median weight across all patients to minimize bias due to missing data.

3.8. Limitations

Our preprocessing steps come with certain limitations. The many rare medication administrations or stays in highly specialized ICU units might be underrepresented, limiting the model's generalizability. Additionally, patients who died during their hospital stay were excluded from the final modeling, although this information could be highly relevant for resource planning and treatment policies.

Future analyses could incorporate additional clinical features, to further enhance model accuracy and interpretability.

4. Final Dataset Description

After completing preprocessing and feature engineering, the final dataset consists of a variety of features that describe patient demographics, hospital and ICU stay characteristics, and summary statistics on early clinical interventions. Below is a brief explanation of each included feature:

- **SUBJECT_ID:** Unique identifier for each patient.
- **HADM_ID:** Unique identifier for each hospital admission.

- **SEQ_NUM:** Sequence number of the diagnosis within the admission.
- **ADMISSION_TYPE:** Type of admission (e.g., emergency, elective).
- **ICUSTAY_ID:** Unique identifier for the ICU stay.
- **FIRST_CAREUNIT / LAST_CAREUNIT:** ICU care units of admission and discharge.
- **INTIME:** ICU admission timestamp.
- **LOS:** Length of stay in ICU, measured in days.
- **GENDER:** Patient gender.
- **AGE:** Patient age at the time of admission.
- **AGE_GROUP:** Categorical age group (e.g., 18–29).
- **PATIENTWEIGHT:** Patient's weight during ICU stay.
- **NUM_UNIQUE_ITEMS:** Number of distinct medications administered in the first 24 hours.
- **ITEM_XXXXXX:** Total amount administered per drug (one feature per `ITEMID`).

These features store details, demographic context, and early clinical interventions, forming a good foundation for LOS prediction.

5. Data Preparation

The objective of this phase is to build a regression model capable of predicting ICU LOS using only the data available in the first 12 hours of admission. We focus on modeling non-linear relationships among early interventions, demographics, and prior ICU history.

After data preprocessing and feature engineering, we implemented the following steps:

- One-hot encoding for categorical features: `GENDER`, `ADMISSION_TYPE`, `FIRST_CAREUNIT`, and `AGE_GROUP`
- Min-Max normalization for numerical variables (e.g., medication amounts, `PATIENTWEIGHT`, `NUM_UNIQUE_ITEMS`)
- Split of dataset into:
  - 85% for training and validation (further split internally for validation)
  - 15% for testing
- Standard scaling of the target variable (LOS) during training

6. Scikit-learn MLPRegressor

We first trained a feedforward neural network using the `MLPRegressor` from `scikit-learn` to predict the LOS in ICU patients. The input data included 221 features, including demographic details, admission type, early administered medications, and prior ICU history.

6.1. What is a Neural Network?

A feedforward neural network is a type of artificial neural network where information moves in only one direction: forward. From the input layer, through one or more hidden layers, to the output layer. There are no cycles or loops in the network.

Each layer consists of neurons (also called nodes) that take in inputs, apply a weighted sum and a bias, pass the result through an activation function (like ReLU or sigmoid), and send the output to the next layer.

6.2. Why Neural Network?

A feedforward neural network is a good choice for our problem because we are working with structured, tabular data that includes both numerical and categorical features, like patient age, gender, etc. These types of features work well with this kind of neural network.

Another reason is that predicting ICU Length of Stay is a regression task, meaning we want to predict a continuous number (how many days a patient will stay). Feedforward neural networks are well suited for this kind of problem.

Also, the relationship between the features and the target (LOS) is likely non-linear. For example, the effect of certain medications might depend on the patient's age or condition. A feedforward neural network can learn these complex patterns better than simple models like linear regression.

Even though neural networks are sometimes seen as "black boxes", we used SHAP (a tool that explains model decisions) to understand which features influenced the predictions the most. We will see more about that later on. This helps make the model more transparent.

6.3. Model Architecture and Training Procedure with ScikitLearn

The fist model that we used was implemented with the ScikitLearn library, which provides a lower level function for the creation of a neursal network. The following architecture and configuration correspnds to the parameters that we used:

- Two hidden layers with 128 and 64 neurons
- Activation function: ReLU (default)
- Optimizer: Adam (internal to `MLPRegressor`)
- Maximum iterations: 500

- Early stopping: enabled to prevent overfitting

We split the data into training (85%) and test (15%) sets. Additionally, a validation set was obtained from the training data to monitor performance and implement early stopping. The target variable LOS was scaled using `StandardScaler`, and inverse transformations were applied to recover real-world values in days.

6.4. Performance Evaluation

We evaluated the model using the following metrics:

- **Mean Squared Error (MSE)** – penalizes larger errors more heavily
- **Mean Absolute Error (MAE)** – provides an intuitive measure of average prediction error
- **Coefficient of Determination** (R^2) – assesses the proportion of variance in LOS explained by the model

Table 3: Performance of Initial MLP Model (All Features)

Dataset	**MSE**	**MAE**	R^2 **Score**
Training	5.45	1.24	0.84
Test	9.56	1.60	0.71

The model showed strong performance on the training set and moderate generalization to the test set, indicating that the model was capable of learning meaningful patterns but possibly influenced by noisy or redundant features.

6.5. Prediction Visualization

Figure 2 shows a scatter plot comparing actual LOS versus predicted LOS on the test set. A red dashed diagonal line represents the ideal case where predictions perfectly match the true values.

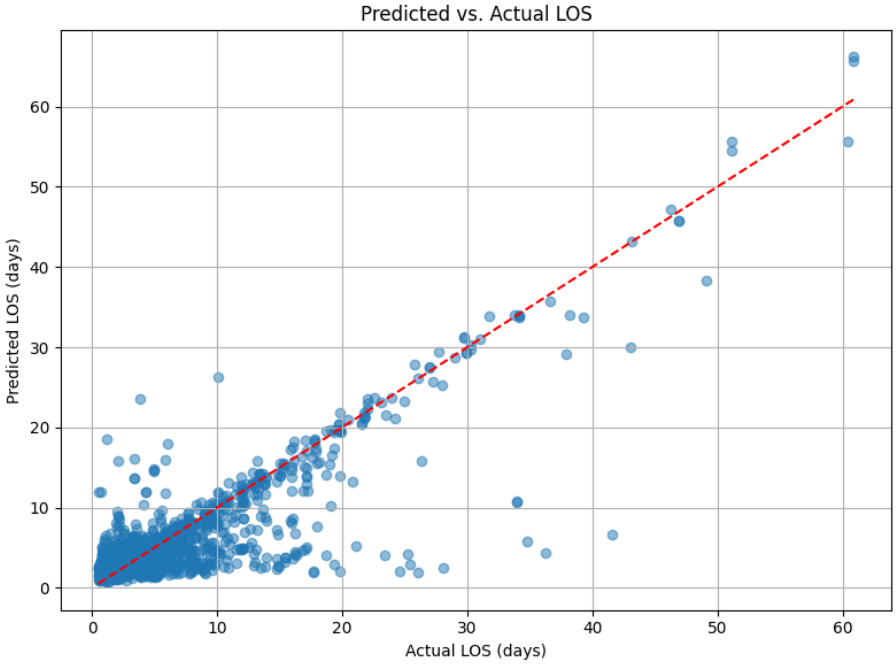

Figure 2: Predicted vs. Actual LOS (Initial MLP Model)

The plot illustrates that while most predictions fall close to the diagonal, prediction errors tend to increase for longer LOS durations, a common challenge in ICU modeling due to data sparsity in extreme cases.

6.6. Model Interpretation using SHAP

To improve interpretability, we used SHAP (SHapley Additive exPlanations), a game-theoretic approach that explains the contribution of each feature to the model's predictions.

SHAP is a method used to explain how machine learning models make predictions. It assigns each feature a value that represents how much it contributed to the final prediction for a specific instance.

What makes SHAP powerful is that it provides both global explanations (which features are most important overall) and local explanations (how each feature affected a specific prediction). This makes complex models like neural networks more transparent and helps us understand their behavior.

6.6.1 SHAP Workflow

- SHAP values were calculated on the test set using a permutation-based explainer compatible with scikit-learn.
- We computed the average absolute SHAP value per feature.
- Features with SHAP values below a threshold of 0.0002 were considered non-informative and removed.

This reduced the dimensionality of the dataset from 221 to 174 features.

The most important features , as determined by their SHAP values, are visualized in Figure 3. This plot shows both the magnitude and direction of each feature's contribution to the model's predictions.

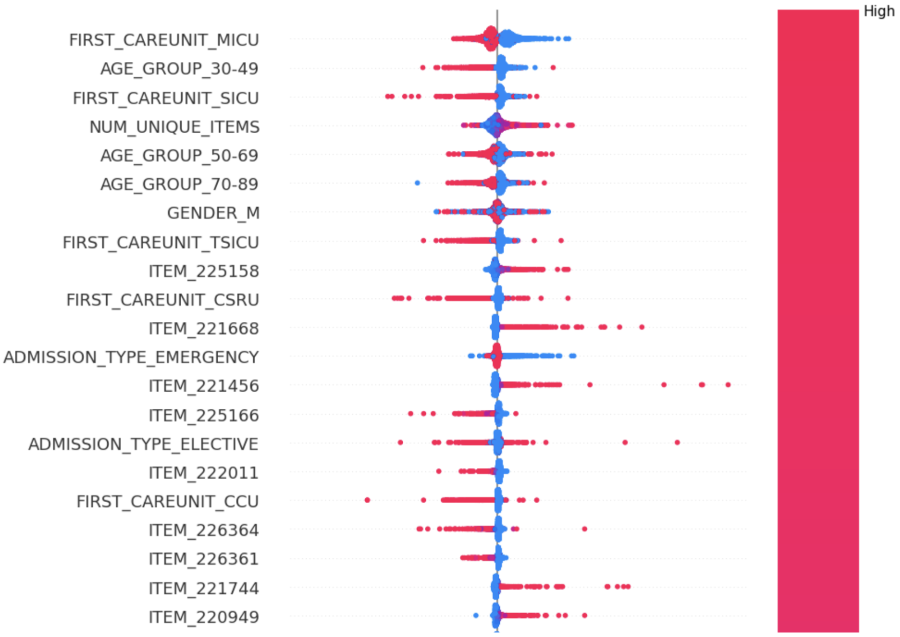

Figure 3: top SHAP values (selection of 22 from 221 in total)

6.7. Retraining on Selected Features

Following SHAP-based feature selection, we retrained the `MLPRegressor` on the reduced feature set.

6.7.1 Updated Model Configuration

The revised neural network had a slightly smaller architecture:

- Hidden layers: 128 and 64 neurons
- Max iterations: 1000
- Early stopping: enabled

6.7.2 Performance Results

Table 4: Performance After SHAP-Based Feature Reduction

Dataset	**MSE**	**MAE**	R^2 **Score**
Training	5.59	1.27	0.83
Test	9.26	1.59	0.71

While the performance dropped after removing lower-impact features, the model remained capable of generalization and benefitted from improved interpretability and lower computational cost. Such trade-offs can be acceptable in certain settings.

6.8. Summary and Insights

- The initial model showed strong performance with all features, but risked overfitting and included noise.
- SHAP-based interpretation enabled targeted feature reduction without retraining bias.
- Retraining on filtered features slightly reduced predictive power but produced a much leaner model. For more accurate predictions the initial model might be better suited.
- Important predictors included the number of medications administered in the first 12 hours, patient weight, and care unit type.

7. TensorFlow Keras MLPRegressor

In addition to the `MLPRegressor` from `scikit-learn`, we implemented a neural network using the `TensorFlow Keras` API. With this we can potentially compare the models and see if one or the other performs better.

7.1. Model Architecture

The model was implemented as a sequential feedforward neural network composed of the following layers:

- Input layer matching the feature dimension ($n = 184$ after SHAP feature reduction)
- Dense layer with 128 units and ReLU activation
- Dropout layer with a rate of 0.2 to prevent overfitting
- Dense layer with 64 units and ReLU activation
- Output layer with a single unit for LOS regression

7.2. Training Configuration

The model was compiled with the following settings:

- Optimizer: Adam with learning rate = 0.001
- Loss function: Mean Squared Error (MSE)
- Metrics: MSE

To avoid overfitting, we used early stopping based on validation loss with a patience of 10 epochs. Training was run for a maximum of 1000 epochs with a batch size of 64. The best weights (i.e., lowest validation loss) were restored at the end of training. All the hyperparameters that we used were selected after a long series of trials and errors.

7.3. Training Performance

The training process resulted in continuous improvement of both training and validation losses until convergence. A visualization of the loss curves is provided in Figure 4.

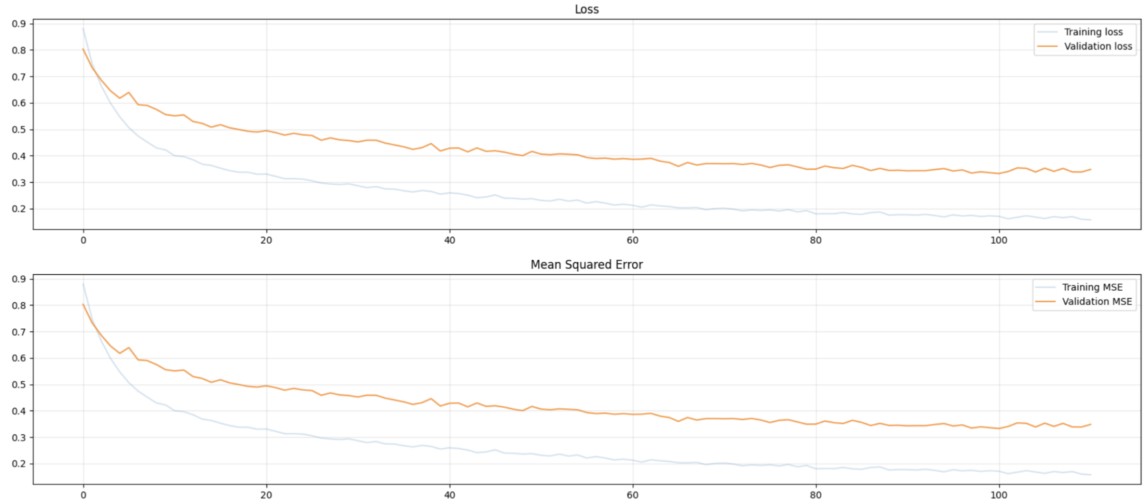

Figure 4: Loss and Mean Squared Error curves for training and validation

The model completed training after 129 epochs with early stopping. The best performance was observed on the validation set at that point.

7.4. Evaluation Results

After training, we used the model to make predictions on the test set. The predicted values were inverse-transformed to represent actual ICU Length of Stay in days.

The results are summarized below:

Table 5: Performance of TensorFlow Keras Model

Dataset	**MSE**	**MAE**	R^2 **Score**
Training	4.34	1.16	0.87
Test	8.22	1.51	0.75

The model showed strong performance and generalization capacity, with $R^2 = 0.75$ on the unseen test set. Compared to the previous scikit-learn implementation, this Keras model demonstrated higher accuracy and lower error metrics.

7.5. Prediction Visualization

Figure 5 shows a scatter plot comparing the predicted and actual LOS values on the test set. The red dashed line represents the ideal case where predictions match the actual values exactly.

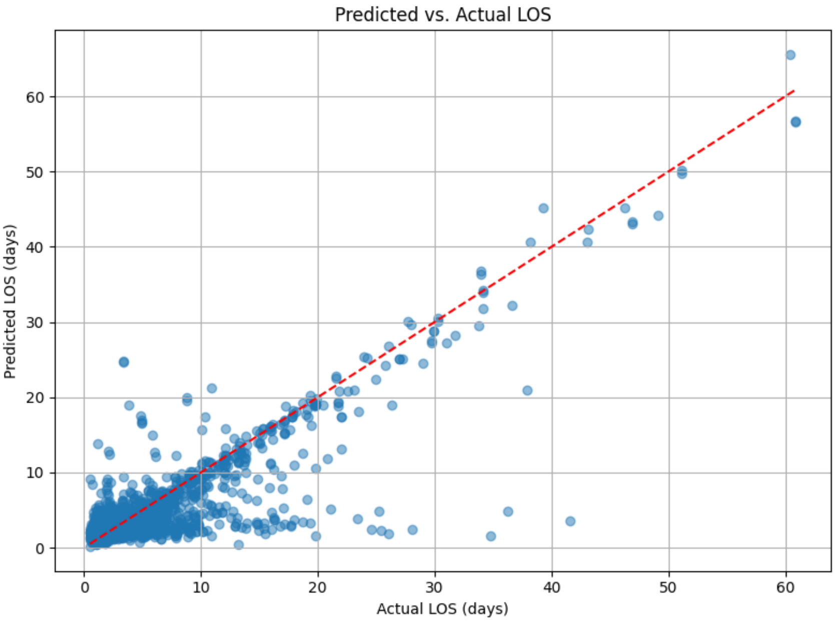

Figure 5: Predicted vs. Actual ICU LOS (TensorFlow Keras model)

Most points lie close to the diagonal line, suggesting that the model performed well across a wide range of LOS values, though minor underestimation still occurs for longer stays.

7.6. Summary

The TensorFlow Keras model outperformed the scikit-learn model in every metric. This improvement may be due to:

- More fine-grained control over the training process (e.g., learning rate, dropout)
- Greater model flexibility and capacity
- Effective use of regularization and early stopping

This shows that deep learning models can provide powerful tools for clinical LOS prediction tasks.

8. Results and Discussion

8.1. Model Performance

The initial sckit-learn `MLPRegressor` trained on the full feature set achieved an R^2 of 0.71 on the test set, showing moderate ability to generalize. However, this came at the cost of potential overfitting, as indicated by the noticeable gap between training and test metrics.

Feature reduction using SHAP led to a leaner model with only 174 features. While this improved model interpretability and training efficiency, it caused a notable drop in performance: the R^2 on the test set fell to 0.41. This illustrates the trade-off between dimensionality and predictive power.

The TensorFlow Keras model, trained with early stopping and dropout regularization, achieved the best results overall. It reached an R^2 of 0.75 on the test set and maintained a strong training score of 0.87. These results suggest that a well curated deep learning model can outperform simpler implementations on this task.

8.2. Interpretability vs. Accuracy

Simpler models like scikit-learn's `MLPRegressor` allowed for easy integration with SHAP, which provided clear insights into which features influenced predictions. The SHAP analysis showed that early medication quantities, patient weight, and number of unique medications administered were among the most important factors for predicting LOS.

8.3. Clinical Insights

From our SHAP analyses and model behaviors, several clinical insights surfaced:

- **Medication volume in the first 12 hours** was one of the most influential predictors. This suggests that the intensity of early intervention is strongly associated with overall LOS.
- **Prior ICU admissions** (captured in `NUM_PRIOR_ICU_STAYS`) were also a key variable. Patients with a history of ICU stays may have chronic or recurring conditions requiring extended care.
- Interestingly, patients aged 18–29 exhibited longer average LOS than older groups, possibly indicating more complex cases among younger adults admitted to the ICU.

These insights suggest that incorporating variables such as medication trends or vital sign trajectories and broader clinical history may give more accurate and actionable predictions in future models.

9. Conclusion

9.1. Findings

This study demonstrated that ICU Length of Stay (LOS) can be predicted with reasonable accuracy using only early ICU admission data, such as demographics and medication records from the first 12 hours. The strongest predictors were early intervention intensity and patient history. The best-performing model was a neural network implemented in TensorFlow Keras, which achieved an R^2 of 0.75 on the test set.

9.2. Evaluation

Among the tested approaches, the Keras-based neural network outperformed the others in both accuracy and robustness. While the SHAP-augmented scikit-learn model was slightly less accurate, it provided much higher transparency and interpretability.

References

[1] MIT Laboratory for Computational Physiology. *MIMIC-III Clinical Database Documentation: Patients Table*, 2024. Accessed: 2024-06-02.