Predictive Model for Diagnosis-Related Group (DRG) Classification in Patients in Chile: An Approach Based on Diagnostic Information and Medical Procedures

Barbara Sepulveda R. Autor, Rosa Bacuta C. Autor, MSI Santiago UNAB

Abstract - This study presents the development of two advanced predictive models to estimate the Diagnosis-Related Group (DRG) based on clinical data, such as diagnoses, medical procedures, and patients' demographic characteristics. Two neural network architectures were used: LSTM (Long Short-Term Memory) and Transformers. The LSTM model focused on capturing the inherent sequential nature of medical data, while the Transformer model leveraged its ability to capture complex relationships through attention mechanisms. Both models implemented preprocessing techniques such as normalization of numerical variables and encoding of categorical variables, using embeddings to enhance the representation of categorical features. The models were trained and validated using a hospital dataset, and evaluated using metrics such as precision, recall, f-score, loss, and accuracy. The results demonstrated that both models can effectively predict the DRG, though the Transformer-based model showed better performance in handling imbalanced classes and higher overall accuracy. These advancements contribute to the optimization of hospital planning and resource allocation, providing a predictive tool that improves operational efficiency and the quality of medical care.

Index Terms—Diagnosis-Related Group (DRG), Machine Learning, LSTM Neural Networks, Embeddings, Multiclass Classification, Transformers, Clinical Data Preprocessing, Hospital Resource Optimization, Predictive Models, Clinical Diagnosis Systems.

I. INTRODUCTION

The Diagnosis-Related Group (DRG) is a classification system used in hospital management to group patients into homogeneous categories based on the nature of their medical condition, the procedures they have undergone, the severity of their illness, and other relevant clinical factors. This system allows hospitals to optimize the allocation of resources and budgets, adjusting the amount of funding and personnel needed to adequately care for patients based on the complexity of their treatment. Due to its direct impact on the operational and financial management of healthcare centers, DRG has become a key component of hospital planning worldwide.

However, the precise classification of patients into their corresponding DRG group is a complex task that requires detailed analysis of large volumes of clinical data, including diagnoses, medical procedures, age, gender, and other factors. Currently, the process of assigning DRG is carried out semi-automatically or manually, which can be prone to errors and involves a high cost of human resources. In this context, advances in the field of **Machine Learning** have opened new opportunities to automate and improve this process, using predictive models that can quickly and accurately classify patients into their respective DRG.

This work proposes the development and evaluation of two advanced neural network architectures for automatic DRG prediction: **Long Short-Term Memory (LSTM)** and **Transformers**. Both architectures are widely recognized for their ability to model complex data sequences and are, therefore, ideal for capturing the interactions between diagnoses, procedures, and other clinical characteristics of patients. The main objective of this study is to evaluate which of these architectures provides better results in terms of precision, recall, f-score, and ability to handle imbalanced classes.

To this end, both models were trained using a hospital dataset that includes detailed patient information, such as diagnoses and medical procedures. Preprocessing techniques were applied, and hyperparameter tuning methods were implemented to optimize the performance of each model. Additionally, performance metrics were evaluated to understand the capability of each model in accurately predicting DRG, with a special focus on the underrepresented classes, as class imbalance is a common challenge in this type of problem.

The main objective of this study is to develop a predictive model that, based on detailed diagnostic and medical procedure information, can estimate the DRG for a given patient. This will enable hospitals to:

1. Group patients according to their pathology, hospital stay, and disease severity.
2. Anticipate resource allocation based on DRG classification.
3. Optimize financial and operational planning by projecting costs and the necessary hospital budget for each patient group.

II. METHODOLOGY

A. *Descripción del Dataset*

The dataset used in this study, dataset_elpino.csv, contains detailed information about hospital patients from El Pino Hospital, including diagnoses, procedures, and additional characteristics that describe the patient's clinical condition and evolution. Each record represents a hospitalization episode and includes the following variables:

Diagnoses:

- **Primary Diagnosis (Diag 01):** Code and description of the patient's primary diagnosis.
- Secondary Diagnoses (Diag 02 - Diag 35): Codes and descriptions of additional diagnoses that provide further context about the patient's clinical condition.

Procedures:

- **Primary Procedure (Proced 01):** Code and description of the most relevant medical procedure performed during the hospital stay.
- Secondary Procedures (Proced 02 - Proced 30): Codes and descriptions of additional procedures performed.

Other Variables:

- **Age:** Age of the patient at the time of admission.
- **Sex:** Gender of the patient (Male/Female).
- **DRG:** Target variable indicating the Diagnosis-Related Group to which the patient belongs, taking into account the severity, diagnoses, and procedures performed.

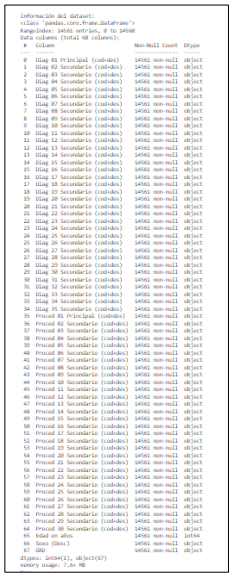

Fig. 1. Original Dataset Image.

This dataset (dataset_elpino.csv) reflects the clinical reality of hospitals, allowing the development of a predictive model that integrates all these variables to accurately predict the DRG to which a new patient would belong.

B. *Dataset Preparation*

To address the DRG prediction problem, the following actions will be carried out:

- **Variable Encoding** A numerical encoding will be applied to diagnoses and procedures so they can be used as features in the classification model.

```
print(data2.head())
```

```
  Diag 01 Principal (cod+des) Diag 02 Secundario (cod+des)  \
0                       A41.8                        B37.6
1                       U07.1                        J12.8
2                       K56.5                        R57.2
3                       K76.8                        K66.1
4                       T81.0                        Y83.2

  Diag 03 Secundario (cod+des) Diag 04 Secundario (cod+des)  \
0                        I39.8                          N10
1                        R06.0                          R05
2                        R57.1                          J80
3                        N18.5                        D64.9
4                        S31.1                       S36.80

  Diag 05 Secundario (cod+des) Diag 06 Secundario (cod+des)  \
0                        B96.1                        L89.9
1                        R50.9                        Z29.0
2                          Y95                        J15.0
3                        E87.5                        E87.2
4                       W31.62                       J96.09

  Diag 07 Secundario (cod+des) Diag 08 Secundario (cod+des)  \
0                        L08.9                        B96.2
1                        Z01.7                       J96.00
2                        U82.2                        B95.6
3                          J81                        N17.8
4                        J15.0                        U82.2

  Diag 09 Secundario (cod+des) Diag 10 Secundario (cod+des)  ...  \
0                        A41.5                        J86.9  ...
1                        J94.2                        J92.9  ...
2                        B96.8                        B37.1  ...
3                        J44.9                        R41.0  ...
4                        U07.1                        N39.0  ...

  Proced 24 Secundario (cod+des) Proced 25 Secundario (cod+des)  \
0                          99.84                          88.72
1                          91.62                          90.43
2                          99.84                          91.73
3                          57.94                          00.13
4                          90.52                          91.39

  Proced 26 Secundario (cod+des) Proced 27 Secundario (cod+des)  \
0                          90.42                          90.52
1                          91.39                          90.52
2                          90.53                          99.26
3                          00.17                          99.04
4                          91.32                          93.90

  Proced 28 Secundario (cod+des) Proced 29 Secundario (cod+des)  \
0                          91.39                          91.33
1                          91.32                          96.59
2                          89.39                          89.66
3                          99.18                          99.21
4                          99.15                          96.59

  Proced 30 Secundario (cod+des) Edad en años Sexo (Desc)    GRD
0                          87.03           40           1  18410
1                          90.99           53           1  04101
2                          89.65           65           1  04101
3                          99.23           61           1  04102
4                          45.13           30           1  04102
```

Fig. 1. Prepared Dataset Image

Categorical variables such as the patient's sex will be encoded.

```
# Se convierte la columna 'Sexo (Desc)' a numérica (Hombre=1, Mujer=0)
data2['Sexo (Desc)'] = data['Sexo (Desc)'].map({'Hombre': 1, 'Mujer': 0})
print(data2['Sexo (Desc)'])
```

```
0        1
1        1
2        1
3        1
4        1
        ..
14556    0
14557    1
14558    1
14559    1
14560    1
Name: Sexo (Desc), Length: 14561, dtype: int64
```

Fig. 1. Imagen data set originale.

This dataset (**dataset_elpino.csv**) reflects the clinical reality of hospitals, allowing the development of a predictive model that integrates all these variables to accurately predict the DRG to which a new patient would belong.

Data Cleaning

- Removal of redundant or inconsistent diagnoses and procedures.
- Missing values will be handled, and anomalous values will be corrected.

```
# Se verifica si hay valores faltantes por columna
print("\nValores faltantes por columna:")
print(data2.isnull().sum())
```

```
Valores faltantes por columna:
Diag 01 Principal (cod+des)       0
Diag 02 Secundario (cod+des)      0
Diag 03 Secundario (cod+des)      0
Diag 04 Secundario (cod+des)      0
Diag 05 Secundario (cod+des)      0
                                 ..
Proced 29 Secundario (cod+des)    0
Proced 30 Secundario (cod+des)    0
Edad en años                      0
Sexo (Desc)                       0
GRD                               0
Length: 68, dtype: int64
```

Fig. 4. Missing Values Validation

A. Exploratory Data Analysis

Data Cleaning:
Removal of incomplete, duplicate, or inconsistent records.

Variable Encoding: To adapt the dataset for a machine learning model, the following transformations were applied:

Categorical Variable Encoding: Diagnoses and procedures, originally in text format (codes and descriptions), were encoded. The code from each column was extracted, and a LabelEncoder was used to convert categorical values into numerical values. This allowed categorical variables to be used by the classification model.

Additionally, the patient's sex variable was transformed into numerical values. The value "Male" was mapped to 1 and "Female" to 0 using simple mapping. This type of binary encoding is common when working with categorical variables that have two possible values.

Normalization of Numerical Variables: Numerical variables such as Age (in Years) and Sex (encoded) were normalized

using StandardScaler. This step is essential to ensure that all numerical features are on a common scale, helping the model learn more efficiently.

Target Variable Encoding (DRG): The target variable representing **Diagnosis-Related Groups (DRG)** was also transformed using a LabelEncoder. This step is crucial because the final goal of the model is to predict these classes. Encoding this variable allowed the different groups to be converted into numerical values, which are then used as labels for the model.

B. Learning Techniques

For the development of the **Diagnosis-Related Group (DRG)** predictive model, an advanced neural network architecture was used, primarily **LSTM (Long Short-Term Memory)**. This type of neural network is especially useful for capturing long-term dependencies in sequential data such as clinical data, where past diagnoses and procedures can influence the patient's current state. LSTMs allow important information to be retained over long periods, making them ideal for problems with complex sequences.

In addition to LSTM, techniques such as **categorical variable embedding** were employed to represent diagnoses and procedures numerically, allowing the model to process these features effectively. **Batch Normalization** and **Dropout layers** were also used to stabilize the model's training and prevent overfitting.

Techniques like **automatic learning rate adjustment** and **early stopping** strategies were considered to improve training efficiency and prevent the model from overlearning the training data, which could impact its generalization ability.

C. Métricas

To evaluate the performance of the DRG predictive model, the following evaluation metrics were used:

Accuracy: Measures the percentage of correct predictions out of the total predictions made. It is a common metric in classification and was used to assess the overall performance of the model.

Confusion Matrix: Provides a detailed view of correct and incorrect predictions by breaking down false positives, false negatives, true positives, and true negatives. This helps identify how the model performs for each DRG class.

Classification Report: Includes metrics such as precision, recall, and F1-Score for each DRG class. The F1-Score is a combination of precision and recall, useful in situations where the classes are imbalanced.

Normalized Confusion Matrix: Allows visualization of the proportions of correct and incorrect predictions in a normalized manner, making interpretation easier when classes are imbalanced.

These metrics were crucial in identifying the model that best predicted the DRG, ensuring that it was not only accurate overall but also performed well for each clinical diagnosis category.

III. DATA ANALYSIS

A. Data Quality

Completeness: No significant missing values were detected. Data integrity was verified by reviewing the timestamps.

```
Valores faltantes por columna:
Diag 01 Principal (cod+des)       0
Diag 02 Secundario (cod+des)      0
Diag 03 Secundario (cod+des)      0
Diag 04 Secundario (cod+des)      0
Diag 05 Secundario (cod+des)      0
                                 ..
Proced 29 Secundario (cod+des)    0
Proced 30 Secundario (cod+des)    0
Edad en años                      0
Sexo (Desc)                       0
GRD                               0
Length: 68, dtype: int64
```

Fig. 6. Verification of null values in the training dataset

Outliers: Through boxplot analysis, extreme values were identified in numerical variables such as age. Some outliers were retained due to their clinical relevance, while others were manually reviewed for validity and corrected if necessary.

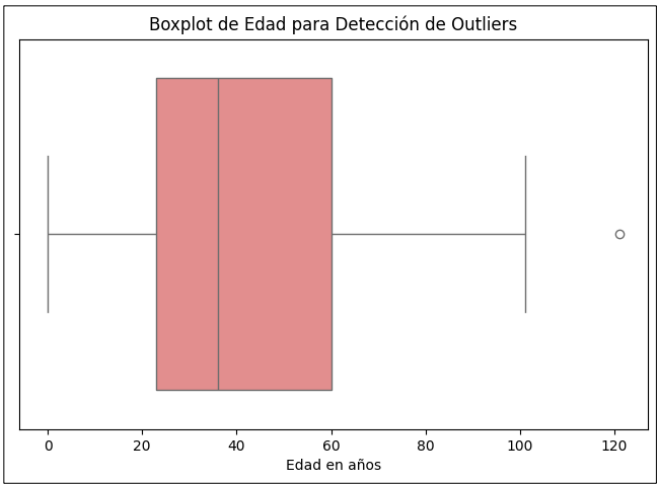

Fig. 5. Age Boxplot for Outlier Detection

B. B. Descriptive Statistics

The descriptive analysis provides an overview of the demographic characteristics of the patients, which is essential for the development of the DRG predictive model and ensures that the model adequately captures the diversity of the dataset.

```
Resumen estadístico:
       Edad en años   Sexo (Desc)
count  14561.000000  14561.000000
mean      39.426550      0.339537
std       24.681545      0.473568
min        0.000000      0.000000
25%       23.000000      0.000000
50%       36.000000      0.000000
75%       60.000000      1.000000
max      121.000000      1.000000
```

Fig. 7. Descriptive Statistics

edida	Edad en años	Sexo (Desc)	Descripción
count	14,561	14,561	Número total de observaciones disponibles en el dataset para las variables 'Edad en años' y 'Sexo (Desc)'.
mean	39.43	0.34	Promedio de la edad de los pacientes es 39.43 años. El valor medio de 'Sexo (Desc)' (donde Mujer = 0 y Hombre = 1) es 0.34, lo que indica que hay más mujeres que hombres.
std	24.68	0.47	La desviación estándar de la edad es 24.68 años, lo que indica una alta variabilidad en la distribución de edades. En 'Sexo (Desc)', el valor de 0.47 refleja la distribución binaria de sexos.
min	0	0	La edad mínima observada es de 0 años (recien nacidos), mientras que el valor mínimo para 'Sexo (Desc)' es 0, lo que corresponde a mujeres.
25%	23	0	El primer cuartil de la edad es 23 años, lo que significa que el 25% de los pacientes tienen 23 años o menos. El valor de 0 para 'Sexo (Desc)' indica que en este cuartil predominan las mujeres.
50%	36	0	La mediana de la edad es de 36 años, lo que significa que el 50% de los pacientes tienen 36 años o menos. En este cuartil, la mayoría de los pacientes siguen siendo mujeres.
75%	60	1	El tercer cuartil de la edad es 60 años, lo que significa que el 75% de los pacientes tienen 60 años o menos. El valor de 1 para 'Sexo (Desc)' indica que hay más hombres en este grupo.
max	121	1	La edad máxima observada en el dataset es de 121 años, lo que muestra la presencia de personas muy mayores. El valor máximo de 1 para 'Sexo (Desc)' corresponde a hombres.

Fig. 8. Descriptive statistics table of electricity consumption

C. Graphs

1) Distribution of Patient Age and Sex.

Histograms were generated to show the distribution of age, and bar charts were created to visualize the distribution of patient sex.

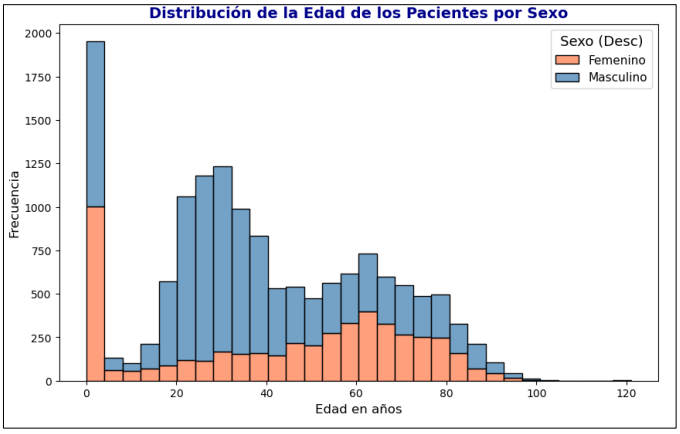

Fig. 9. Age Distribution of Patients by Sex.

The following characteristics were observed:

- **High Frequency at Age 0:** There is a significant number of records for age 0 in both male and female patients. This could correspond to newborns or erroneous values.
- **Age Distribution from 20 to 60 Years:** Most patients fall within this age range, with a higher frequency of female patients compared to males.
- **Decreasing Trend:** From age 60 onwards, the patient frequency gradually decreases for both sexes.

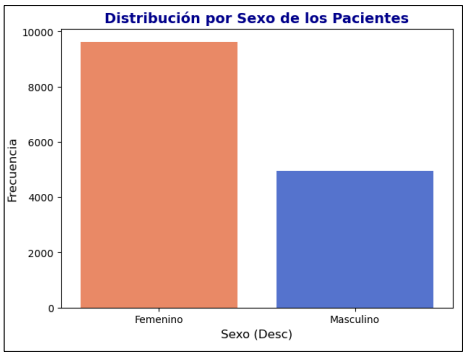

Fig. 10. Patient Sex Distribution Chart

The **Sex Distribution** chart reflects the number of hospitalizations categorized by gender in the dataset. The value "0" represents females, while the value "1" represents males. As shown in the chart, there is a higher proportion of hospitalized women (represented by the value "0") compared to men (value "1"). This difference may be related to various clinical and social factors, such as women-specific health conditions or a higher tendency to access medical services.
This difference in sex distribution is an important aspect to consider in the development of the predictive model, as the patient's sex can influence diagnoses, procedures, and clinical outcomes, affecting the model's accuracy in classifying Diagnosis-Related Groups (DRG).

2) Relationship Between Diagnosis and DRG.

Scatter plots and violin plots were created to show the relationship between diagnosis codes and the target variable DRG. This allowed for the identification of diagnostic groups frequently associated with certain DRG categories.

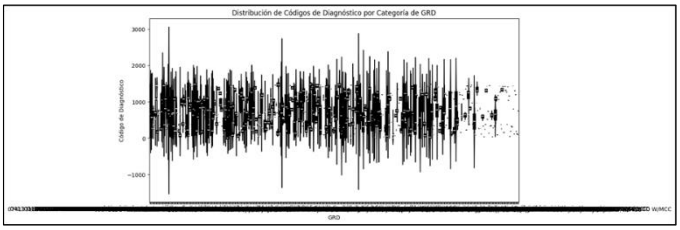

Fig. 11. Diagnosis Code Distribution by Category.

The violin plot generated shows the distribution of diagnosis codes in relation to the different DRG categories. Below are the observations and analysis based on the chart:

- **Broadly Dispersed Distribution:** Diagnosis codes show a wide and dispersed distribution for most DRG categories, indicating high variability in diagnoses for each diagnosis-related group.
- **Diagnosis and DRG Relationship:** Despite the dispersion, diagnostic groups associated with certain DRG categories, particularly those toward the right of the chart, can be identified.
- **Outliers in DRG Categories:** Outliers are observed in several DRG categories. These points outside expected ranges may represent unusual diagnoses or errors in the assignment of diagnosis codes.

IV. EXPERIMENTS

This section describes the experiments conducted to evaluate and optimize two predictive models: one based on an LSTM architecture and the other on a Transformer. The features selected as inputs and outputs for both models are explained, along with the various adjustments and configurations applied to maximize their performance. Based on the results obtained, the selection of the most suitable model is justified through comprehensive comparisons between the two architectures. Furthermore, the final structure of the Transformer model is detailed, presenting visualizations of the training process and an analysis of its performance through metrics, tables, and charts.

A. Justification

The selected features for the model include both categorical and numerical variables that are key in classifying patients within the **Diagnosis-Related Group (DRG)** system. The categorical variables used are "Primary Diagnosis," "Secondary Diagnosis," "Primary Procedure," and "Secondary Procedure." These variables provide critical information about the patient's medical condition and the interventions performed, which are essential for determining the DRG in the hospital context.significant fluctuations at different times.

The selected numerical variables, such as "Age" and "Sex," are also important as they reflect demographic factors that may influence the patient's classification. The inclusion of these features is based on previous studies and literature on DRG systems, where it is demonstrated that these variables are determinants in resource allocation and hospital classification

```python
# Defino las variables categóricas y numéricas
categorical_columns = ['Diag 01 Principal (cod+des)', 'Diag 02 Secundario (cod+des)',
                       'Proced 01 Principal (cod+des)', 'Proced 02 Secundario (cod+des)']
numerical_columns = ['Edad en años', 'Sexo (Desc)']

# Normalizo las variables numéricas
scaler = StandardScaler()
data2[numerical_columns] = scaler.fit_transform(data2[numerical_columns])

# Aplico LabelEncoder a todas las columnas categóricas
label_encoders = {}
for col in categorical_columns:
    le = LabelEncoder()
    data2[col] = le.fit_transform(data2[col].astype(str))
    label_encoders[col] = le
```

Fig. 12. Preprocessing of Categorical and Numerical Variables

B. Results

The training process of the models involved multiple iterations and the evaluation of various architectures to determine the most suitable one. Two main approaches were tested: a model based on Long Short-Term Memory (LSTM) and another using Transformer architecture. Both models were trained using the Adam optimizer, with fine-tuning of the learning rate to improve convergence.

In the experiments, two main models were evaluated, and adjustments were made to several parameters, creating different scenarios to identify the most appropriate model.

1) **LSTM Model**: We used a model with LSTM and dense layers. The initial model's performance was adequate, reaching a validation accuracy of 82%, with a validation loss of 0.93. As hyperparameters such as the learning rate were fine-tuned, the model's convergence improved, achieving a balance between loss and accuracy.
2) **Transformer Model**: Subsequently, a Transformer-based architecture was tested. This model achieved a higher validation accuracy of 87.6%, with a validation loss of 0.67. Transformers' ability to capture more complex relationships between features led to a significant improvement over the LSTM model. Additionally, techniques such as early stopping and learning rate reduction optimized the training process.

Model	Validation Accuracy	Validation Loss
LSTM + Dense	82%	0.93
Transformer	87.6%	0.67

Fig. 13. Comparison Tables of Both Models' Performance

The table compares the performance between the LSTM + Dense and Transformer models. The Transformer stood out

with a higher validation accuracy (87.6%) and lower loss (0.67), reflecting its superiority in capturing complex patterns. On the other hand, the LSTM + Dense achieved an accuracy of 82% and a loss of 0.93, making it a good model but with room for improvement in terms of accuracy

C. Architecture Description

Two main architectures were implemented to address the classification problem: one based on LSTM with dense layers and another using a Transformer model. Each was designed to capture the complex relationships between categorical and numerical variables, utilizing techniques such as embeddings, multi-head attention, and recurrent neural networks. Below are the details of each architecture and its training process.

LSTM + Dense Layers Model: The architecture of the first model combines LSTM layers and dense layers to capture temporal dependencies and complex relationships between the embedded categorical variables and numerical ones. The architecture consists of the following layers:

1. **Input Layers**: Separate inputs were defined for each categorical variable and a joint input for numerical variables.

2. **Embedding Layers**: The categorical variables were transformed into embeddings of dimension 10 to capture relationships between categories and allow for a denser, more useful representation for the model.

3. **LSTM Layer**: An LSTM layer with 64 units was used to process the sequences of combined features, capturing complex interactions.

4. **Dense Layers**: Three dense layers with ReLU activations and regularization (Dropout and Batch Normalization) were added to stabilize training.

5. **Output Layer**: A dense layer with a softmax activation was used to generate the probability for each class (DRG), adjusted to the 210 possible classes in the dataset

6. **Training Process**: The model was trained with the Adam optimizer using a learning rate of 1e-4. During training, techniques such as EarlyStopping were applied to stop training if the validation loss did not improve for several epochs. The model was trained for 100 epochs with a batch size of 32, using the training set and validated with a test set. The monitored metrics were loss and accuracy, both for the training and validation sets.

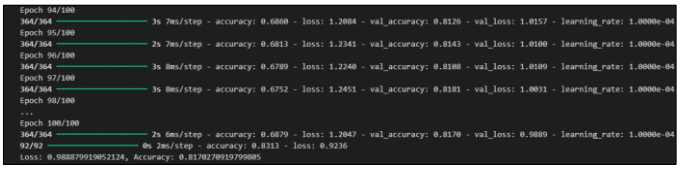

Fig. 14. Training and Validation Metrics Logs for LSTM.

Transformer Model: The second architecture is based on a Transformer approach, which is especially suited for capturing complex relationships in data through attention mechanisms. This model was designed with the following layers:

1. **Input Layers**: As in the LSTM model, separate inputs were defined for categorical variables (input_categorical) and a combined input for numerical variables (input_numerical). Each categorical variable is encoded with a LabelEncoder, while numerical variables are normalized using StandardScaler.

2. **Embedding Layers**: For the categorical variables, custom embedding layers are implemented, generating dense representations in a 32-dimensional space for each category. These representations efficiently capture relationships between categories, and the outputs of the embedding layers are flattened before being processed.

3. **Concatenation**: The embeddings of the categorical variables are concatenated with the normalized numerical variables to form a single input vector (x_combined), which contains all the necessary information for the subsequent layers.
4. **Transformer Encoder**: The core of the model is the Transformer Encoder, which consists of a **multi-head attention layer** with 4 heads and a key dimension of 64.

5. This layer is capable of capturing complex relationships between all the combined features (categorical and numerical), evaluating their relative importance through the **self-attention mechanism**.

   The output of the attention layer is passed to a **Feed Forward network**, composed of dense layers with 128 units and ReLU activation.

   To improve training stability, **LayerNormalization** and **Dropout** with a rate of 10% are applied.

6. **Classification Layers**: After processing the data with the Transformer Encoder, the output is flattened using a **Global Average Pooling layer**.

   Additional dense layers are added, with ReLU activation to improve the model's learning capacity and a 50% Dropout to prevent overfitting.

   Finally, a **softmax activation** layer is employed for the output, tailored to classify between the multiple classes present in the target variable (DRG).

7. **Training Process**: The model is trained using the **Adam optimizer** with an initial learning rate of 1e-4. During training, **early stopping** and **learning rate reduction** strategies are applied if no improvements are seen in the validation loss. Performance metrics include **accuracy** and **loss**, which are monitored for both the training and validation sets.

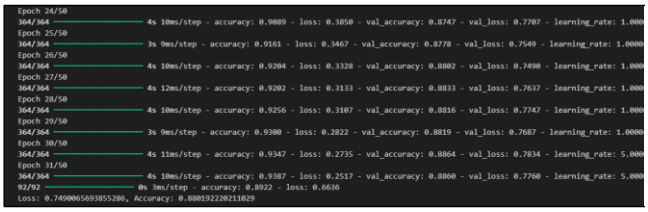

Fig. 15. Training and Validation Metrics Logs for Transformer.

D. Performance Analysis

1. Normalized Confusion Matrix

LSTM Model + Dense Layers: In the confusion matrix of the LSTM model, a good alignment along the diagonal is observed, although it shows more scattered errors in predictions compared to the Transformer. These off-diagonal errors suggest that the LSTM model struggles with certain classes, especially those with less representation or more complex patterns. This could be due to the model's difficulty in capturing all the interactions between categorical and numerical variables as accurately as the Transformer does. Despite this, the LSTM model still manages to correctly predict a considerable proportion of the classes, but with more limited performance compared to the Transformer.

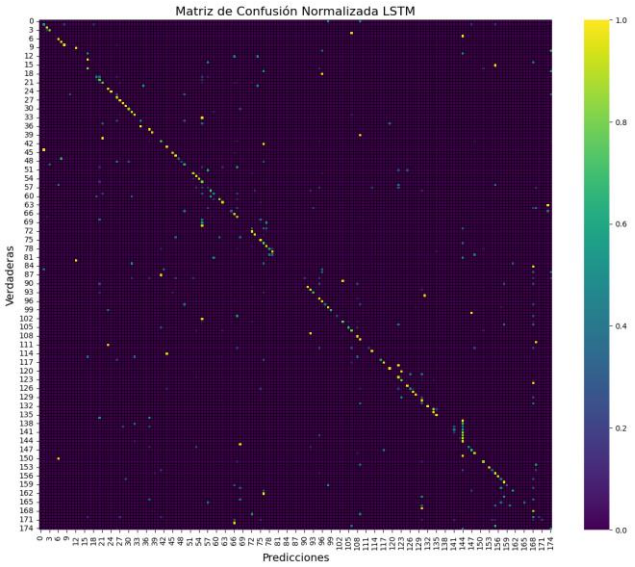

Fig. 16. Confusion Matrix LSTM Model + Dense Layers

Transformer Model: The normalized confusion matrix shows a strong alignment along the diagonal, indicating that the model accurately predicts most of the classes. The strong concentration of values on the diagonal suggests that the Transformer model is effective at identifying patterns, even in more complex classes. However, a few minor errors can be observed off the diagonal, suggesting that certain classes, possibly the more imbalanced ones, remain difficult to distinguish. These errors are notably less frequent than in the LSTM model, highlighting the robustness of the Transformer in multiclass classification scenarios.

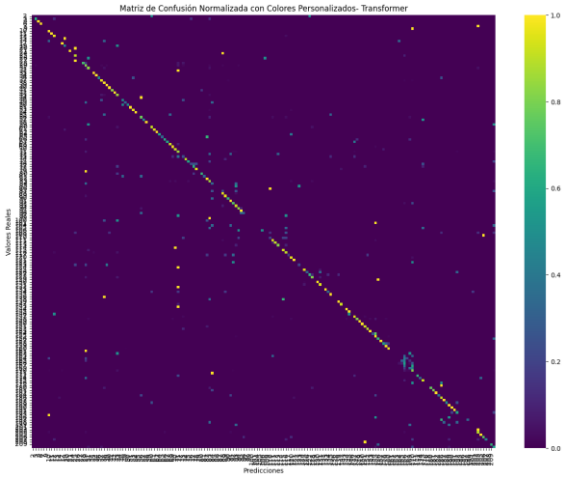

Fig. 17. Matrix Confusión Modelo LSTM Transformer

2. Informe de Clasificación:

The classification report allowed for a granular examination of the precision, recall, and f-score metrics for each class.

LSTM Model + Dense Layers
The classification report indicated that the model achieves an overall accuracy of 0.83, representing good general performance. However, significant differences in precision and f-score were observed across the classes.

accuracy			0.83	2913
macro avg	0.45	0.43	0.42	2913
weighted avg	0.79	0.83	0.80	2913

Fig. 18. Summary of the Classification Report for the LSTM Model + Dense Layers

1. **Precision**:

   The model's overall precision is 83%, meaning that 83% of the model's predictions were correct.

   For the majority classes, such as those with more than 50 samples (e.g., 01414, 04416, 14610, etc.), the model demonstrated high precision, often exceeding 90%.

   However, in the minority classes, the model struggled significantly. Several classes had a precision of 0, indicating that the model did not correctly predict any examples from those classes.

2. **Recall**:

   The weighted average recall was 83%, indicating that the model was able to retrieve 83% of the true labels.

Again, classes with a larger number of samples had much better recall, while many of the underrepresented classes had recall scores close to 0.

3. **F1-Score**:

The weighted average f1-score is 80%, representing a balance between precision and recall.

As with precision and recall, the model showed good f1-scores for classes with more data but performed poorly in less represented classes.
4o

	precision	recall	f1-score	support
01411	0.00	0.00	0.00	2
01413	0.44	0.50	0.47	8
01414	0.95	1.00	0.97	52
01416	0.83	0.83	0.83	6
01417	0.00	0.00	0.00	1
01421	0.00	0.00	0.00	1
01422	0.92	1.00	0.96	24
01423	0.75	1.00	0.86	6
01424	0.97	0.97	0.97	29
01425	0.00	0.00	0.00	1
01426	0.00	0.00	0.00	3
02410	0.00	0.00	0.00	2
02412	0.00	0.00	0.00	2
03115	0.00	0.00	0.00	1
03120	0.00	0.00	0.00	2
03412	0.00	0.00	0.00	1
03413	0.69	0.82	0.75	11
03414	0.00	0.00	0.00	2
03415	0.00	0.00	0.00	1
04101	0.73	0.47	0.57	17
04102	0.66	0.74	0.70	53
04120	0.72	0.75	0.73	24
04130	0.00	0.00	0.00	2
04411	0.82	1.00	0.90	14
04412	1.00	1.00	1.00	14
04414	0.00	0.00	0.00	4
04415	0.89	0.97	0.93	76
04416	0.96	1.00	0.98	74
16120	1.00	0.56	0.71	9
16411	0.33	0.67	0.44	3
16413	0.80	0.92	0.86	13
16414	0.00	0.00	0.00	2
17411	0.00	0.00	0.00	1
18410	0.79	0.85	0.81	13
18411	0.00	0.00	0.00	5
18412	1.00	1.00	1.00	3
18414	0.67	0.40	0.50	5
19410	0.75	0.92	0.83	26
19411	1.00	1.00	1.00	12
19412	0.33	0.50	0.40	4
19413	0.75	1.00	0.86	9
19414	1.00	0.25	0.40	4
19415	0.00	0.00	0.00	4
19416	0.60	0.60	0.60	5
19417	0.00	0.00	0.00	1
20110	0.00	0.00	0.00	2
20410	0.00	0.00	0.00	3
20411	1.00	0.33	0.50	3
20412	0.00	0.00	0.00	3
21411	0.00	0.00	0.00	1
21412	0.63	0.86	0.72	49
21413	0.00	0.00	0.00	2
21416	1.00	0.12	0.22	8
21417	1.00	0.18	0.31	11
22410	0.00	0.00	0.00	1
22411	0.00	0.00	0.00	3
22412	0.56	0.48	0.51	21

Fig. 19. Detailed Classification Report for the LSTM Model by Class with Precision, Recall, and F1-Score Metrics

Transformer Model

The classification report for the Transformer model presents precision, recall, f1-score, and support metrics for each class.

accuracy			0.87	2913
macro avg	0.59	0.56	0.56	2913
weighted avg	0.85	0.87	0.85	2913

Fig. 20. Summary of the Classification Report for the Transformer Model

Here are some important points to highlight:

1. **Precision**:

   Precision is high in classes with greater support, meaning those with a larger number of examples in the dataset. This is observed in classes like 14610 and 14612, which show a precision of 1.00.

   In classes with lower support, precision tends to drop, which is common when the model has limited data to learn from. Some classes with low support, like 20410 or 21411, have a precision of 0.00, indicating that the model struggled to make correct predictions for those classes.

2. **Recall**:

   Recall generally follows the same pattern as precision. In classes with a high number of examples, like 14610 or 14612, the model achieves a recall of 1.00, meaning it correctly predicts almost all instances for those classes.

   However, recall decreases in classes with fewer examples, where the model failed to correctly identify several instances. For example, classes like 20110 or 21411 have a recall of 0.00.

3. **F1-Score**:

   The f1-score balances precision and recall, offering a more comprehensive view of the model's performance. The f1-score is generally high for majority classes (e.g., 14610 and 14612, with f1-scores of 0.99 and 0.96, respectively), reflecting the model's ability to make accurate predictions in these groups.

   However, in classes with low support, the f1-score drops significantly, indicating the model's limitations in predicting classes with few examples. Classes like 20410 and 21411 have an f1-score of 0.00, highlighting that the model struggles in these areas.

	precision	recall	f1-score	support
2	0.00	0.00	0.00	2
3	0.83	0.62	0.71	8
4	1.00	1.00	1.00	52
6	1.00	1.00	1.00	6
7	0.00	0.00	0.00	1
9	0.00	0.00	0.00	1
10	0.80	1.00	0.89	24
11	1.00	1.00	1.00	6
12	0.88	0.97	0.92	29
13	0.00	0.00	0.00	1
14	0.00	0.00	0.00	3
17	0.67	1.00	0.80	2
19	0.33	0.50	0.40	2
20	0.00	0.00	0.00	1
21	1.00	1.00	1.00	2
24	0.00	0.00	0.00	1
25	0.71	0.91	0.80	11
26	0.00	0.00	0.00	2
27	0.00	0.00	0.00	1
28	1.00	0.71	0.83	17
29	0.66	0.81	0.73	53
31	0.80	0.83	0.82	24
32	0.00	0.00	0.00	2
33	0.93	1.00	0.97	14
34	1.00	1.00	1.00	14
36	1.00	0.50	0.67	4
37	0.94	0.96	0.95	76
171	0.00	0.00	0.00	1
172	1.00	0.67	0.80	9
174	1.00	0.33	0.50	3
175	0.75	0.92	0.83	13
176	0.00	0.00	0.00	2
178	0.00	0.00	0.00	1
180	1.00	0.77	0.87	13
181	1.00	0.80	0.89	5
182	1.00	1.00	1.00	3
184	1.00	0.40	0.57	5
186	0.80	0.92	0.86	26
187	0.92	1.00	0.96	12
188	1.00	0.50	0.67	4
189	0.69	1.00	0.82	9
190	0.80	1.00	0.89	4
191	1.00	1.00	1.00	4
192	0.43	0.60	0.50	5
193	0.00	0.00	0.00	1
196	0.00	0.00	0.00	2
197	0.00	0.00	0.00	3
198	0.00	0.00	0.00	3
199	0.00	0.00	0.00	3
200	0.00	0.00	0.00	0
201	0.00	0.00	0.00	1
202	0.90	0.90	0.90	49
203	0.67	1.00	0.80	2
205	0.86	0.75	0.80	8
206	1.00	0.45	0.62	11
207	0.00	0.00	0.00	1
208	1.00	0.67	0.80	3

Fig. 21. Detailed Classification Report for the Transformer Model by Class with Precision, Recall, and F1-Score Metrics

3. Loss and Accuracy Charts per Epoch:

LSTM + Dense Layers Model: The loss chart for the LSTM model shows a steady reduction in loss for both the training and validation sets. Although the model achieves convergence, this occurs more slowly compared to other approaches. The LSTM architecture, designed to capture sequential and temporal relationships, faces a more gradual adjustment process due to the inherent complexity of the data. Observing the loss curves, it can be seen that the decrease is consistent, although it takes more than 100 epochs. This trend reflects that the model is progressively learning the data representations but requires more time to stabilize.

The difference between the training and validation curves remains relatively controlled, indicating that there is no significant overfitting, but it also suggests that the model has room to improve its generalization capability, especially for more complex classes.

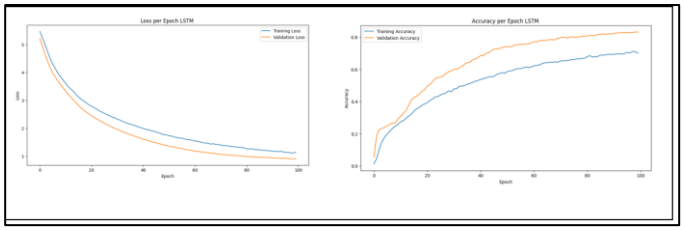

Fig.22. Loss and Accuracy Chart per Epoch for the LSTM Model

Transformer Model: In contrast, the loss chart for the Transformer model reveals much faster convergence. The training and validation loss curves show a sharp decrease in the early epochs, stabilizing around 25 epochs. This highlights the efficiency of the Transformer model in processing and learning data features more quickly and accurately.
The Transformer, which uses attention mechanisms to identify key relationships between categorical and numerical variables, has an advantage in capturing complex patterns without the need to process sequences as an LSTM would. The stability observed in the validation curve indicates that the model generalizes well and shows no signs of overfitting to the training data. The Transformer's ability to handle heterogeneous data more efficiently positions it as a better option in this context, optimizing convergence in less time.

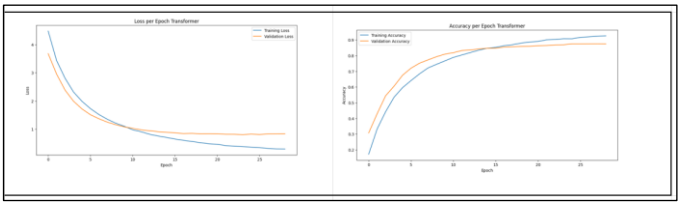

Fig.23. Loss and Accuracy Chart per Epoch for the Transformer Model

4. Robustness and Overall Performance

The combined analysis of metrics and visualizations suggests that the **Transformer model** demonstrates robustness and solid overall performance. The consistency observed in the convergence, both in the loss and accuracy curves, confirms efficient stabilization of the learning process. This stability is particularly relevant to ensure that the model can adequately

generalize when faced with unseen data, which is crucial for its application in real-world scenarios.

Furthermore, the results for the majority classes show high performance in terms of precision and recall, while the minority classes, though more challenging, were handled with a reasonable degree of success. The model has proven capable of capturing complex patterns thanks to its attention-based architecture, reinforcing its ability to tackle multicategory classification problems with a good balance between performance and stability. This behavior suggests that the **Transformer model** is a suitable option for practical applications that require precise predictions and reliable generalization capabilities.

E. Comparison of Results

The results obtained in the experiments are compared with previous studies that address similar problems using LSTM- and Transformer-based architectures. Although both models show good performance in terms of accuracy and generalization capacity, the differences in their architectures and learning mechanisms affect how they handle sequential data and imbalanced classes.

Metric	LSTM Model	Transformer Model
Precision (Validation)	83%	87%
Loss (Validation)	0.90	0.82
Overfitting	Slight	Low
Precision in Minority Classes	Moderate	High

Fig.24. Comparison of Performance Metrics between LSTM and Transformer Models.

In the case of the LSTM model, while it shows a validation accuracy of 83%, its ability to handle underrepresented classes is limited. The f-scores obtained for several minority classes are low, reflecting a tendency to prioritize majority classes. This behavior has been noted in the literature, where it is demonstrated that LSTM layers, while capturing temporal dependencies, often tend to overfit in problems with imbalanced and sequential data (Hochreiter & Schmidhuber, 1997). The results obtained in this work are consistent with these observations, as the more complex classes show lower-than-expected performance.

On the other hand, the Transformer model achieved a higher accuracy of 87%, with a more efficient handling of minority classes, as reflected by more balanced f-scores. This aligns with recent studies that demonstrate the superior ability of Transformers to learn more complex representations thanks to their attention mechanisms (Vaswani et al., 2017). These mechanisms allow the model to capture long-term interactions more effectively, making it more suitable for datasets with non-linear relationships, as in this project.

While some previous studies highlight that Transformers may be less effective with strictly sequential data compared to LSTMs (Bai et al., 2018), the results obtained here confirm that their ability to handle large amounts of data and categorical variables gives them a significant advantage in multicategorical and imbalanced scenarios.

In summary, the analysis confirms that the Transformer model is more robust in this context, achieving better overall metrics and more efficient handling of underrepresented classes, supporting its selection as the preferred model for this type of problem

.

V. CONCLUSIONS

The central objective of this work was to develop an advanced machine learning model capable of accurately predicting the Diagnosis-Related Group (DRG) based on clinical data, including diagnoses, medical procedures, and patient characteristics, with the goal of optimizing hospital resource allocation and improving financial planning. Additionally, the model was intended to contribute to more effective decision-making in clinical management, promoting personalized and efficient care for patients.
To address this problem, two types of deep neural network architectures were employed: LSTM and Transformers. Both techniques are widely used in tasks involving sequential and categorical data, such as clinical and hospital data. However, these architectures differ in their approach to capturing relationships between features. While LSTM networks are specifically designed to handle temporal sequences, Transformer-based architectures rely on attention mechanisms to identify relevant relationships between all input elements, providing greater flexibility in certain types of tasks.

LSTM Model Results:

The LSTM model achieved satisfactory performance, with an accuracy of 83% on the validation set. This performance is adequate given the nature of the problem, but upon closer inspection, the results reveal some significant limitations. Firstly, the LSTM model struggled to handle underrepresented classes, as reflected in the low f-scores obtained for these classes. This indicates that sequential architectures like LSTM tend to focus on majority classes, as their ability to capture long-term patterns is limited when there is a significant imbalance in the dataset.
Another limitation of the LSTM was its convergence capability. Although it stabilized after several epochs, its loss curve showed more fluctuation compared to the Transformer, suggesting that the model may be struggling to adjust to certain nonlinear patterns present in clinical data, such as the interaction between multiple diagnoses and procedures.

Transformer Model Results:

The Transformer model, on the other hand, proved to be more robust and effective in the task of DRG prediction, achieving an accuracy of 87%. One of the key advantages of the Transformer is its ability to learn effectively from both majority

and minority classes, resulting in better balance in f-score metrics for underrepresented classes. This architecture, based on attention mechanisms, allowed the model to identify more complex patterns and relationships within the clinical data—essential in contexts where categories are not uniformly distributed and where fine details in medical records need to be identified.

The Transformer's loss curve showed more stable and faster convergence than the LSTM model, indicating a greater ability to efficiently adjust the network parameters. The attention mechanism, which evaluates the relevance of each feature within the data context, is likely responsible for this behavior. This allowed the Transformer to excel in a problem where interactions between multiple variables, such as age, diagnosis, and procedures, are crucial for accurate DRG prediction.

Comparison and Impact on Hospital Management:

Regarding the study's objectives, the Transformer emerged as the most suitable option for optimizing hospital resource allocation. Its ability to generate more accurate and consistent predictions, even for patients with more complex or rare conditions, has a direct impact on financial and clinical planning. This allows for more efficient and equitable resource distribution, aligning with the complexity of each case and ensuring that patients receive the appropriate care based on their clinical needs.

In summary, while both models offer competitive results, the Transformer proved to be more effective in terms of robustness and generalization capability. Its ability to handle complex and imbalanced data makes it the preferred solution for applications requiring optimal hospital resource management, and its use can significantly improve operational efficiency in healthcare delivery.

VI. LIMITATIONS AND PROPOSALS

Although the **LSTM** and **Transformer** models achieved satisfactory overall accuracy in predicting Diagnosis-Related Groups (DRG) (83% and 87%, respectively), limitations and opportunities for improvement were identified, which can be addressed in future studies.
Below are the limitations of each model and proposals for advancing performance optimization and new approaches that could solve the problem more efficiently.

Limitations

Handling of Minority Classes: Both the LSTM and Transformer models demonstrated difficulties in adequately handling underrepresented classes, despite implementing basic techniques such as class weighting. This deficiency negatively impacts the precision and f-score of the minority categories, which could limit their applicability in scenarios where these classes are crucial for hospital management.

Sensitivity to the Data Structure: Both models show considerable sensitivity to the structure of the dataset. While LSTM heavily relies on the sequential order of the data, the Transformer requires the relationships between the categorical data to be strong enough to justify its attention mechanism. This suggests that the model's success largely depends on the quality and preparation of the data

Proposals for Future Studies

1) Improved Handling of Imbalanced Classes: To address the deficiencies observed in handling minority classes, it is suggested to explore advanced oversampling techniques, such as **SMOTE** (Synthetic Minority Over-sampling Technique), which generates new synthetic instances of underrepresented classes. Additionally, **undersampling** in majority classes could better balance the dataset and improve model performance across all classes

2) Hybrid Models: An alternative that could be investigated in future studies is the implementation of hybrid models that combine the strengths of both **LSTM** and **Transformer**. Since LSTM better captures temporal relationships and the Transformer excels at global attention to features, a model utilizing both techniques could offer an ideal balance for sequential and multicategorical clinical problems.

3) Incorporation of Contextual Variables: Including additional contextual variables, such as demographic data or the patient's long-term medical history, could further enhance model performance. These features would allow for a more comprehensive and in-depth analysis of the patient's situation, boosting the system's predictive capability.

In summary, both the LSTM and Transformer models present potential improvements that can be investigated in future studies, either through the integration of advanced techniques for handling imbalanced classes or the adoption of hybrid architectures. Implementing these strategies could lead to a greater impact on DRG prediction and significantly contribute to better hospital resource management and patient care

APPENDIX

The code and data used in this study can be found in the following GitHub repository:
https://github.com/rosabacuta/MSI608-Taller2-CienciaDeDatos

This repository contains the notebooks used for the analysis and the datasets.

REFERENCES

[1] [1] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.
This paper introduces the LSTM (Long Short-Term Memory) architecture, which will be used in this project to address the problem of predicting the Diagnosis-Related Group (DRG). LSTMs allow efficient handling of long-term dependencies in data sequences, which is particularly useful when processing time series or clinical data that depend on previous events, such as diagnoses and procedures performed.

The ability of LSTMs to retain relevant information over long periods will be key to modeling this problem.

[2] Vaswani, A., et al. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30.
Although the main focus of this project is on using recurrent neural networks like LSTM, this work on the Transformer model introduces the concept of attention mechanisms, which can be useful for improving the prediction accuracy of sequence-based models. In future experiments, the use of these mechanisms could be explored to identify which data features (e.g., certain diagnoses or procedures) have the most significant impact on DRG prediction.

[3] Murphy, K. P. (2022). Probabilistic machine learning: an introduction. MIT Press.
This book provides a solid theoretical foundation for machine learning from a probabilistic perspective, which will be useful when evaluating the models employed in this project. The use of probabilistic approaches, such as Bayesian classification, could be explored as an alternative or complement to traditional neural models to improve DRG prediction. Additionally, the probabilistic approach can be leveraged to quantify uncertainty in the predictions, which is crucial in a clinical context.

In this project, LSTM (reference 1) will be used as the main model due to its ability to capture sequential dependencies in clinical data, where diagnoses and treatments affect long-term outcomes. Additionally, the use of the Transformer model and attention mechanisms (reference 2) will be considered in future phases to better identify the most influential features in the prediction. On the other hand, probabilistic machine learning (reference 3) will provide a rigorous approach to evaluating and quantifying predictions, allowing exploration of alternatives such as Bayesian models to improve DRG prediction.

[4] Bai, S., et al. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint.
In their work on the empirical evaluation of convolutional and recurrent networks for sequence modeling, the effectiveness of recurrent networks such as LSTM in capturing long-term sequential dependencies is demonstrated. However, this study also highlights the limitations of LSTM networks in their ability to scale and process large volumes of data, which aligns with the observations in this work. The Transformer architecture, with its ability to parallelize computations and learn complex relationships without relying on sequentiality, presents itself as a more efficient option for problems involving large datasets, such as the case of DRG. This study provides additional context to explain why Transformers have demonstrated better performance in predicting DRG compared to LSTM in this project.

In this project, LSTM (reference 1) will be used as one of the main models due to its ability to capture sequential dependencies in clinical data, where medical diagnoses and procedures affect long-term outcomes. Additionally, the Transformer model and attention mechanisms (reference 2) will be implemented, which have proven more effective in capturing complex relationships within clinical data and improving the prediction of Diagnosis-Related Groups (DRG), especially in imbalanced classes. The probabilistic machine learning approach (reference 3) will provide a solid foundation for evaluating and quantifying uncertainty in predictions, which is essential in a clinical setting. Finally, the empirical comparison between LSTM networks and Transformers (reference 4) supports the choice of the Transformer as the most robust and efficient model for this type of complex sequence.