IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication Nov 21, 2025, date of current version Nov 21, 2025.

Digital Object Identifier 10.1109/ACCESS.2024.XXXXX

Prediction of Diagnosis-Related Groups (DRG) using Machine Learning

CAMILA EYZAGUIRRE[1], CRISTIAN LORCA[1], CINTYA OLIVARES[1], and ALEJANDRO SUAREZ[1]

[1]Universidad Andrés Bello, Santiago, Chile (e-mail: c.eyzaguirrevillarro@uandresbello.edu, c.lorcavargasvargas@uandresbello.edu, c.olivarescisternas@uandresbello.edu, a.suarezsantelices@uandresbello.edu)

Corresponding author: Camila Eyzaguirre (e-mail: c.eyzaguirrevillarro@uandresbello.edu).

ABSTRACT Accurate prediction of Diagnosis-Related Groups (DRG) is fundamental for efficient hospital resource management and medical care planning. This study presents a machine learning model based on XGBoost to predict DRG codes from clinical information, including diagnoses, procedures, and patient demographic data. The dataset used contains 14,561 records with 68 features, including primary and secondary diagnoses, procedures, and demographic variables. A balancing strategy was implemented through grouping of minority classes and using sample weights to address the extreme imbalance observed (proportions of up to 1218:1 between classes). The model was trained with 205 features derived from the top 100 most frequent diagnoses and procedures, along with demographic variables and aggregated characteristics. An accuracy of 59.22% was obtained in predicting 166 grouped DRG categories, significantly improving over a random model. The results demonstrate the feasibility of using machine learning techniques for automatic DRG prediction, contributing to the optimization of hospital processes.

INDEX TERMS Machine learning, DRG, medical prediction, XGBoost, multi-class classification, imbalanced data

I. INTRODUCTION

DIAGNOSIS-Related Groups (DRG) constitute a patient classification system widely used in hospital management and health resource planning [1], [2]. These codes group patients with similar clinical characteristics and resource consumption, facilitating hospital efficiency comparison and resource allocation. Accurate and early prediction of DRG codes represents a significant challenge in the healthcare field, as these codes depend on multiple clinical factors, including primary and secondary diagnoses, performed procedures, patient age, and other demographic factors.

The traditional process of assigning DRG codes is manual and requires specialized clinical expertise, which can result in inconsistencies, delays in coding, and suboptimal resource use. Automating this process through machine learning techniques offers the promise of improving accuracy, reducing processing time, and providing more consistent predictions [3], [4].

This work addresses the problem of DRG code prediction as a multi-class classification task, where each DRG code represents a different class. The main challenge lies in the extreme class imbalance, where some codes appear hundreds of times while others appear only once in the dataset. This imbalance poses significant problems for machine learning algorithms, which tend to favor majority classes at the expense of minority classes [5].

II. LITERATURE REVIEW

The application of machine learning techniques in predicting medical classification codes has been widely explored. Several studies have applied algorithms such as Random Forest, Support Vector Machines (SVM), and neural networks for the classification of diagnoses and procedure codes [6], [7].

Chen and Guestrin introduced XGBoost as an efficient and scalable implementation of gradient boosting, demonstrating excellent performance in multi-class classification problems [8]. In the medical context, XGBoost has shown promising results due to its ability to handle heterogeneous data, missing values, and imbalanced classes through the use of sample weights [9].

The handling of imbalanced classes in medical classification problems has been addressed through various strategies, including undersampling, oversampling, SMOTE, and the use of class weights [10]. However, in cases of extreme imbalance, grouping of minority classes has proven to be an effective strategy that preserves information without introducing artificial biases [11].

Appropriate metrics for evaluating models in imbalanced problems include not only accuracy, but also precision, recall, F1-score, and especially multi-class logarithmic loss (mlogloss), which penalizes errors in probability estimation [12].

III. OBJECTIVES

The main objective of this study is to develop a machine learning model capable of predicting DRG codes from clinical and demographic information of patients. Specific objectives include:

1) Analyze the quality and distribution of available data to identify relevant characteristics and detect imbalance problems.
2) Implement a preprocessing strategy that adequately handles extreme class imbalance through grouping and sample weights.
3) Develop a multi-class classification model using XGBoost optimized for the DRG prediction problem.
4) Evaluate model performance using appropriate metrics for imbalanced multi-class problems.

IV. METHODOLOGY

A. DATASET DESCRIPTION

The dataset used in this study contains 14,561 patient records, each characterized by 68 variables. The main features include:

- **Diagnoses**: One primary diagnosis and up to 29 secondary diagnoses, each with its code and description.
- **Procedures**: Multiple performed procedures, each with code and description.
- **Target variable**: DRG code that includes the base code plus a severity digit (MCC, CC, or without CC).
- **Demographic data**: Patient age, sex, and other demographic characteristics.

The dataset initially contained 14,561 records, but after removing duplicates, all unique records were retained for analysis. The target variable (DRG) presents extreme imbalance, with codes appearing up to 1,218 times and others appearing only once.

B. METHODOLOGY OVERVIEW

The model development process followed a structured methodology that included the following stages:

1) Data Preprocessing

Data preprocessing consisted of:

1) **DRG code cleaning**: The base code was extracted by removing the last digit that represents severity (MCC/CC), resulting in 5-digit codes.
2) **Diagnosis and procedure cleaning**: Only codes were extracted, removing textual descriptions.
3) **Duplicate removal**: Duplicate records were identified and removed.
4) **Minority class grouping**: DRG codes with frequency less than 5 were grouped into a category "OTHER_DRG", reducing classes from 210 to 166.

2) Feature Engineering

205 features were created through:

- **Binary diagnosis features**: 100 features indicating presence/absence of the top 100 most frequent diagnosis codes.
- **Binary procedure features**: 100 features indicating presence/absence of the top 100 most frequent procedure codes.
- **Demographic features**: Age (numerical), sex (binary encoded as two variables).
- **Aggregated features**: Total number of diagnoses and total number of procedures per patient.

This strategy allowed capturing the most relevant information while keeping the problem dimensionality manageable.

3) Machine Learning Techniques

a: XGBoost Selection and Justification

XGBoost (eXtreme Gradient Boosting) was selected as the main algorithm for the following reasons:

- **Handling imbalanced classes**: XGBoost allows assigning sample weights that balance the relative importance of minority and majority classes during training.
- **Native multi-class classification**: Directly supports multi-class classification problems through the *multi:softprob* objective function, avoiding the need for one-vs-rest or one-vs-one strategies.
- **Missing value handling**: XGBoost automatically handles missing values in data, which is common in clinical data.
- **Interpretability**: Provides feature importance measures, facilitating understanding of which factors most influence predictions.
- **Performance**: Has demonstrated excellent performance in machine learning competitions and similar problems in the medical domain.
- **Built-in regularization**: Includes L1 and L2 regularization parameters that help prevent overfitting, crucial when the number of features is large.

b: Training Configuration

The XGBoost model was configured with the following hyperparameters:

- **objective**: *multi:softprob* for multi-class classification with probabilities.
- **eval_metric**: *mlogloss* (multi-class logarithmic loss).
- **max_depth**: 6 tree depth levels.
- **learning_rate**: 0.1 for balanced learning.
- **subsample**: 0.8 (row subsampling for regularization).
- **colsample_bytree**: 0.8 (column subsampling).

- **min_child_weight**: 3 (overfitting control in leaves).
- **reg_alpha**: 0.1 (L1 regularization).
- **reg_lambda**: 1.0 (L2 regularization).
- **n_estimators**: 200 (maximum number of trees).
- **early_stopping_rounds**: 20 (early stopping if no improvement).

Validation during training was used with an evaluation set that monitored both the training and test sets, allowing real-time overfitting detection.

The complete project code, including the Jupyter notebook with all preprocessing, training, and model evaluation stages, is publicly available at: https://github.com/Camiev/UNAB-MSI606-202581.1980/blob/main/proyecto1.ipynb

4) Evaluation Metrics

a: Metric Selection and Justification

For model evaluation in this imbalanced multi-class classification problem, the following metrics were selected:

1) **Accuracy**: Proportion of correct predictions over the total. Although it can be misleading in imbalanced problems, it provides a general measure of model performance.
2) **Multi-class Logarithmic Loss (mlogloss)**: Main metric used during training and evaluation. This metric penalizes not only incorrect classifications, but also unreliable probability estimates. It is especially relevant because:
   - It evaluates the quality of predicted probabilities, not just the predicted class.
   - It penalizes more errors in classes with incorrect high confidence.
   - It is appropriate for multi-class problems with multiple classes.
   - It provides information about model calibration.

   The mlogloss formula for a multi-class problem is:

   $$L_{log} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{M} y_{i,c}\log(p_{i,c}) \tag{1}$$

   where N is the number of samples, M is the number of classes, $y_{i,c}$ is 1 if sample i belongs to class c, and $p_{i,c}$ is the predicted probability that sample i belongs to class c.
3) **Classification Report**: Includes precision, recall, and F1-score per class, providing a detailed view of performance in each DRG category.
4) **Confusion Matrix**: For the most frequent classes, it allows visualizing classification errors and understanding which DRG codes are confused with each other.

These metrics were selected because, together, they provide a comprehensive model evaluation: mlogloss for optimization and calibration, accuracy for general interpretability, and precision/recall for detailed per-class analysis.

V. EXPERIMENTS AND RESULTS

A. DATA ANALYSIS

1) Data Quality Assessment

a: Completeness

The completeness analysis revealed that initially no null values were found in the original dataset, indicating good data quality in terms of completeness. However, after code processing, some derived variables may contain missing values, which were handled through specific strategies during feature creation.

b: Correctness

Correctness verification included:

- Validation of DRG code consistency before and after cleaning.
- Verification of coherence between diagnoses and procedures.
- Identification of possible coding errors through analysis of anomalous frequencies.

It was identified that the code structure was consistent and followed expected formats.

c: Outliers

The outlier analysis focused on the distribution of DRG codes. The following were identified:

- **Codes with extremely low frequency**: 73 codes with frequency less than 10, of which 45 had frequency less than 5, including several codes that appeared only once.
- **Codes with extremely high frequency**: 37 codes with frequency greater than 100, with code 14610 being the most frequent with 1,218 occurrences.

These extreme values do not necessarily represent errors, but rather the heterogeneous nature of medical conditions and procedures. However, they require special handling strategies for machine learning.

2) Descriptive Statistics

The statistical analysis of DRG code distribution revealed:

- **Total unique codes**: 210 distinct DRG codes after cleaning.
- **Frequency distribution**:
  - Mean: 68.65 occurrences per code
  - Median: 19 occurrences
  - Standard deviation: 141.05 (indicating high variability)
  - Minimum: 1 occurrence
  - Maximum: 1,218 occurrences
- **Quartiles**:
  - Q1 (25%): 6 occurrences
  - Q2 (Median, 50%): 19 occurrences
  - Q3 (75%): 71.5 occurrences

These statistics confirm the extreme imbalance of the dataset. The median (19) is significantly lower than the mean (68.65), indicating a highly right-skewed distribution, typical of data with imbalanced classes.

a: Top 10 Most Frequent DRG Codes

The most frequent DRG codes represent approximately 40% of all records:

1) 14610: 1,218 occurrences (8.37%)
2) 14612: 925 occurrences (6.35%)
3) 14613: 741 occurrences (5.09%)
4) 07114: 501 occurrences (3.44%)
5) 13416: 458 occurrences (3.15%)
6) 11412: 357 occurrences (2.45%)
7) 04415: 341 occurrences (2.34%)
8) 06120: 332 occurrences (2.28%)
9) 06113: 326 occurrences (2.24%)
10) 04416: 319 occurrences (2.19%)

3) Visualizations

Visualizations were generated to better understand the data distribution:

- **Horizontal bar chart**: Shows the top 20 most frequent DRG codes, allowing identification of dominant categories.
- **Frequency histogram**: Visualizes the complete distribution of DRG code frequencies, showing concentration in low values and presence of outliers.
- **Boxplot**: Illustrates dispersion and outliers in the frequency distribution.
- **Range bar chart**: Categorizes codes according to frequency ranges (less than 10, between 10 and 100, greater than 100), facilitating understanding of imbalance.

These visualizations visually confirmed the extreme imbalance and justified the need for special balancing strategies.

B. PRELIMINARY MODEL RESULTS

1) Data Splitting

The dataset was divided into training and test sets with the following characteristics:

- **Training set size**: 11,533 samples (80%)
- **Test set size**: 2,884 samples (20%)
- **Splitting strategy**: Stratified by class, ensuring that the proportion of each class remains similar in both sets.
- **Classes in training**: 166 unique classes (after grouping)

2) Class Balancing Strategy

A dual balancing strategy was implemented:

1) **Minority class grouping**: DRG codes with frequency less than 5 were grouped into the "OTHER_DRG" category, reducing classes from 210 to 166.
2) **Sample weights**: Weights were calculated using *compute_sample_weight* with 'balanced' strategy, which assigns weights inversely proportional to class frequency.

Examples of weights assigned to the first 10 classes:

- Class 0: weight 11.58 (frequency: 6)
- Class 1: weight 2.48 (frequency: 28)
- Class 2: weight 0.40 (frequency: 173)
- Class 4: weight 17.37 (frequency: 4)

These weights reflect the principle that minority classes receive greater importance during training.

3) Training Process

The model was trained with the following configuration parameters:

- **Features used**: 205 features (100 diagnoses + 100 procedures + 3 demographic + 2 aggregated)
- **Classes to predict**: 166 DRG categories
- **Unique diagnoses in dataset**: 3,649
- **Unique procedures in dataset**: 904

The training process showed constant improvement in the mlogloss metric:

- **Iteration 0**: mlogloss (train) = 4.572, mlogloss (test) = 4.594
- **Iteration 50**: mlogloss (train) = 1.022, mlogloss (test) = 1.734
- **Iteration 100**: mlogloss (train) = 0.639, mlogloss (test) = 1.596
- **Best iteration (130)**: mlogloss (train) = 0.537, mlogloss (test) = 1.588

The model reached its best performance on the test set at iteration 130, after which slight overfitting was observed (improvement in training but deterioration in test).

4) Model Performance

The final model obtained the following results:

- **Accuracy**: 59.22%
- **Best mlogloss (test)**: 1.588 (iteration 130)
- **Improvement over baseline**: The initial mlogloss of 4.594 was reduced to 1.588, representing a 65.4% improvement.

For a multi-class classification problem with 166 classes, an accuracy of 59.22% represents significantly better performance than a random classifier (which would achieve approximately 0.6% accuracy). The mlogloss of 1.588 indicates that the model provides reasonably calibrated probability estimates.

5) Analysis of Results

The preliminary results demonstrate that:

1) **Approach viability**: It is possible to predict DRG codes with reasonable accuracy using machine learning techniques.
2) **Balancing effectiveness**: The grouping strategy combined with sample weights allowed handling extreme imbalance without losing significant information.
3) **Feature selection**: The approach of using the top 100 most frequent diagnoses and procedures captured the most relevant information while maintaining manageable dimensionality.
4) **Room for improvement**: The gap between training mlogloss (0.537) and test mlogloss (1.588) suggests

moderate overfitting, indicating opportunities to improve regularization.

VI. DISCUSSION

The developed model demonstrates that automatic prediction of DRG codes is viable using modern machine learning techniques. The 59.22% accuracy is promising considering the complexity of the problem (166 classes) and the extreme data imbalance.

The minority class grouping strategy proved effective, preserving information from rare classes without introducing artificial biases that could arise from extreme oversampling or undersampling techniques.

The use of XGBoost with sample weights allowed the model to learn from all classes, including minority ones, which is crucial for a medical prediction system that must be useful for all conditions, not just the most common ones.

However, areas for improvement were identified:

- **Regularization**: Adjusting regularization hyperparameters could reduce overfitting and improve generalization.
- **Feature engineering**: Exploring feature interactions or temporal characteristics could improve performance.
- **Ensemble methods**: Combining multiple models could increase robustness and accuracy.

VII. CONCLUSION

This study presents a machine learning model for DRG code prediction that:

- Effectively handles extreme imbalance through grouping and sample weights.
- Achieves 59.22% accuracy in a multi-class classification problem with 166 classes.
- Uses 205 features derived from diagnoses, procedures, and demographic data.
- Demonstrates the feasibility of automating DRG code assignment.

The preliminary results are promising and suggest that with additional refinements, this approach could become a valuable tool for hospital management. Future work should explore hyperparameter optimization, advanced feature engineering techniques, and validation on additional datasets.

CODE AVAILABILITY

The complete source code of this project, including the Jupyter notebook with all preprocessing, feature engineering, XGBoost model training, and evaluation stages, is publicly available on GitHub: https://github.com/Camiev/UNAB-MSI606-202581.1980/blob/main/proyecto1.ipynb

REFERENCES

[1] R. B. Fetter, Y. Shin, J. L. Freeman, R. F. Averill, and J. D. Thompson, “Case mix definition by diagnosis-related groups,” Med. Care, vol. 18, no. 2, pp. 1–53, 1980.

[2] J. P. Weiner, I. Dobson, S. L. Maxwell, K. Coleman, B. Starfield, and G. Anderson, “Risk-adjusted Medicare capitation rates using ambulatory and inpatient diagnoses,” Health Care Financ. Rev., vol. 18, no. 3, pp. 77–99, 1996.

[3] K. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Marcus, M. Sun, P. Sundberg, H. Yee, K. Zhang, Y. Zhang, G. Flores, G. E. Dahl, M. Furst, S. A. Kuhlmann, J. Hughes, J. B. Patil, W. A. Chou, K. de Fauw, J. R. Ledsam, O. Ronneberger, “Scalable and accurate deep learning with electronic health records,” NPJ Digit. Med., vol. 1, no. 1, pp. 1–10, 2018, doi: 10.1038/s41746-018-0029-1.

[4] Z. Obermeyer and E. J. Emanuel, “Predicting the future—big data, machine learning, and clinical medicine,” N. Engl. J. Med., vol. 375, no. 13, pp. 1216–1219, 2016, doi: 10.1056/NEJMp1606181.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: synthetic minority over-sampling technique,” J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.

[6] S. Liu, B. Liu, H. Elhajj, W. Fang, X. Liu, and H. Yu, “Multiclass classification of mechanical ventilated ICU patients by ICU outcome using machine learning,” in Proc. IEEE Int. Conf. Bioinformatics Biomed. (BIBM), Madrid, Spain, 2018, pp. 570–575, doi: 10.1109/BIBM.2018.8621175.

[7] J. Heaton, N. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.

[8] T. Chen and C. Guestrin, “XGBoost: a scalable tree boosting system,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, San Francisco, CA, USA, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[9] J. H. Friedman, “Greedy function approximation: a gradient boosting machine,” Ann. Statist., vol. 29, no. 5, pp. 1189–1232, 2001, doi: 10.1214/aos/1013203451.

[10] M. A. Mazurowski, P. A. Habas, J. M. Zurada, J. Y. Lo, J. A. Baker, and G. D. Tourassi, “Training neural network classifiers for medical decision making: the effects of imbalanced datasets on classification performance,” Neural Netw., vol. 21, no. 2-3, pp. 427–436, 2008, doi: 10.1016/j.neunet.2007.12.031.

[11] Y. Sun, A. K. Wong, and M. S. Kamel, “Classification of imbalanced data: a review,” Int. J. Pattern Recognit. Artif. Intell., vol. 23, no. 4, pp. 687–719, 2009, doi: 10.1142/S0218001409007326.

[12] D. M. W. Powers, “Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation,” J. Mach. Learn. Technol., vol. 2, no. 1, pp. 37–63, 2011.