Patient Mortality Prediction Report

Group 1a Members: Natalie, Hanane, and Austin
HIDS 6001: Massive Health Data Fundamentals

Introduction

Healthcare institutions aim to integrate data-driven models and predictive algorithms into clinical care and diagnosis. The purpose of this shift is to optimize clinical decision-making and improve patient outcomes by leveraging insights from retrospective data analysis [1]. To align with this aim, the EHR Dream Challenge tasked participants with predicting patient mortality within 180 days of their last recorded medical exam. Therefore, we developed an end-to-end machine learning pipeline designed to predict patient morality within the specified timeframe.

Before working with the data, we first defined our true positives and true negatives. Specifically, a true positive is a patient who passed away within 180 days of their last visit. A true negative was defined as a patient who did not pass away within 180 days or passed away after 180 days. For this task we used a synthetic dataset called the Synthetic Public Use File (Synpuf), created by Centers for Medicare and Medicaid Services. The data is organized according to the OMOP Common Data Model to simulate realistic claims data while ensuring the protection of patient privacy.

Methodology

Data Preprocessing

We utilized Python libraries, including pandas for data manipulation and numpy for numerical operations, to preprocess the data. First, we loaded the training and testing datasets, which comprised multiple CSV files representing various EHR components (e.g., person, visit_occurrence, condition_occurrence, drug_exposure, and death tables). Next, we merged these tables with their corresponding concept tables to include human-readable descriptions. To improve data quality, we removed columns with excessive missing values and excluded invalid or redundant entries during the cleaning process.

Feature Engineering

We extracted and transformed relevant features into a structured format for model training. The table below summarizes the key features:

Feature Group	Features Extracted	Processing
Demographics	Age, Gender, Race	- Calculated age from birth_date and death_date. - Encoded gender and race as binary variables using one-hot encoding (drop_first=True, dummy_na=True).
Conditions	Myocardial Infarction, Congestive Heart Failure, Cerebrovascular Disease, Peripheral Vascular Disease	- Grouped conditions using keywords from the Charlson Comorbidity Index [2]. - Transformed into binary features (1 = condition present, 0 = absent).
High-Risk Admissions	ICU, hospice, and palliative care admissions	- Identified using observation data. - Created binary flags by searching for specific keywords in concept_name.
Medications	Cancer drugs, palliative care drugs, serious disease drugs	- Categorized drug exposures using predefined lists. - Created binary flags for exposure to relevant drug categories.

Table 1. Summary of features extracted, including demographics, conditions, high-risk admissions, and medications, with corresponding processing steps for feature extraction and engineering.

Outcome Variable

We derived the target variable (classification) using the following logic:

- **Positive Case (1)**: Patients who passed away within 180 days of their last recorded visit.
- **Negative Case (0)**: Patients who either passed away after 180 days, or Were alive at the time of data cutoff.

To create the target variable, we first Identified Death Dates and used the death table to determine mortality outcomes. Then Filtered Visits and retained only the most recent visit for each patient from the visit_occurrence table. These two columns allowed us to calculate the number of days between the last visit and death date (if available) then assigned labels based on whether the difference was ≤180 days (1) or >180 days (0).

Final Feature Sets

Based on the extracted and processed features, we created three distinct feature groups for model training:

Feature Group 1: Demographics Only

Feature group 1 includes basic demographic features such as age, gender, and race. These features were chosen to assess the power of demographic variables in predicting patient morality.

Feature Group 2: Demographics + Conditions
Feature group 2 combines demographic features with binary-encoded health conditions, including Myocardial Infarction, Congestive Heart Failure, Cerebrovascular Disease, and Peripheral Vascular Disease. This group evaluates whether adding specific comorbidities improves model performance.

Feature Group 3: Full Feature Set

Feature Group 3 is a comprehensive feature set that includes demographics, conditions, high-risk admission flags (e.g., ICU, hospice, and palliative care), and medication categories (e.g., cancer drugs, palliative care drugs). This group assesses the cumulative impact of all extracted features on predictive performance.

Models

To predict patient mortality within 180 days of their last recorded visit, two machine learning models were implemented: Logistic Regression and Random Forest Classifier.

Logistic Regression was configured with the parameter `class_weight='balanced'` to address the severe class imbalance present in the dataset. This adjustment assigns higher weights to the minority class (positive cases), helping the model better account for the underrepresented class and reduce bias towards the majority class.

Random Forest Classifier was configured with `class_weight='balanced_subsample'`, which adjusts the weights at the individual tree level during training. This setting helps mitigate the effects of class imbalance by recalibrating the class weights for each bootstrap sample used to build the trees.

The dataset exhibited a significant class imbalance, with very few positive cases, making these weight adjustments essential to prevent the models from being biased towards predicting the majority class.

Evaluation Metric
The primary evaluation metric chosen for assessing model performance was Area Under the Receiver Operating Characteristic Curve (**AUC-ROC**), as it provides a robust measure of a model's ability to discriminate between classes, even in the presence of class imbalance. This metric is particularly useful because it is less sensitive to imbalanced class distributions compared to other metrics like Precision and F1-Score, which may be disproportionately low due to the dominance of false positives in an imbalanced dataset.

Results

Below are the key evaluation metrics for the two models across the three feature groups:

Model	Precision	Recall	F1-Score	AUC-ROC
Feature Group 1: Demographics				
Logistic Regression	0.01	0.59	0.02	0.683
Random Forest	0.01	0.35	0.02	0.606
Feature Group 2: Demographics + Conditions				
Logistic Regression	0.01	0.53	0.02	0.662
Random Forest	0.01	0.24	0.02	0.563
Feature Group 3: Full Feature Set				

Logistic Regression	0.01	0.53	0.02	0.657
Random Forest	0.01	0.16	0.02	0.542

Table 2. Performance metrics for Logistic Regression and Random Forest across three feature groups.

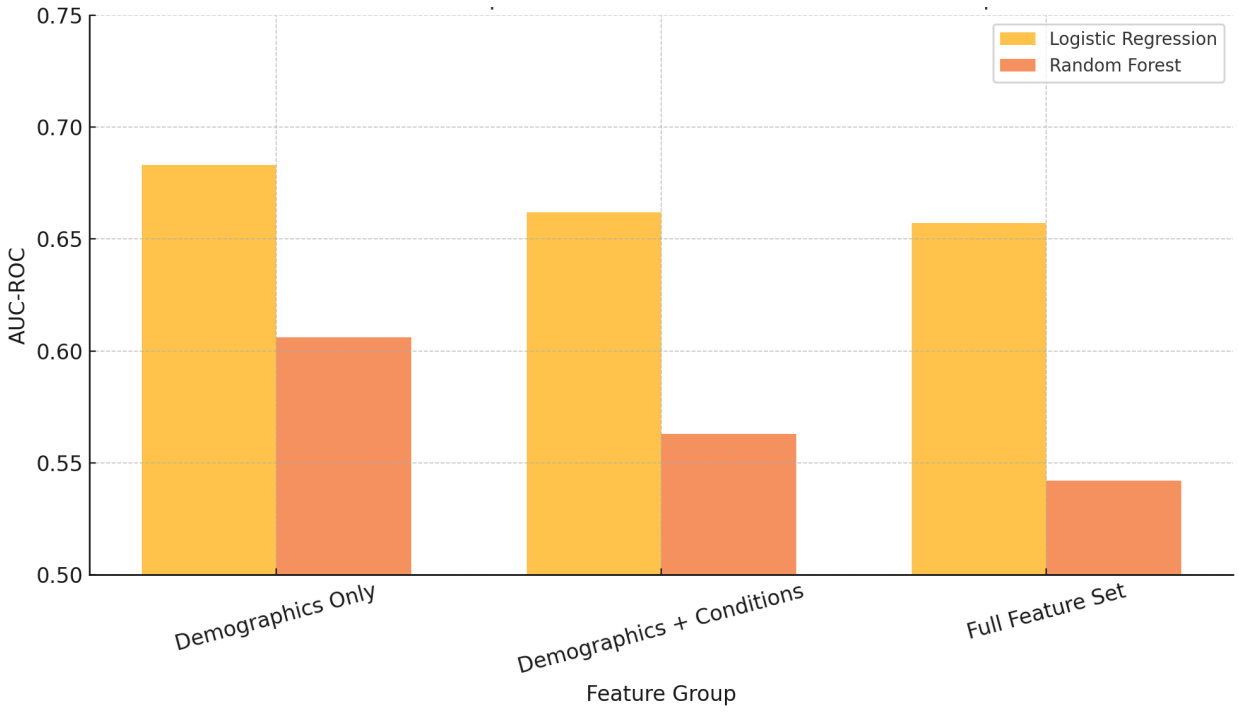

Figure 1. AUC-ROC comparison across models and feature groups.

Analysis

All models demonstrated very low precision (~0.01), which reflected a high rate of false positives. This issue was likely driven by the significant class imbalance in the dataset, making it challenging for the models to accurately identify true positive cases without overpredicting.

In terms of recall, Logistic Regression outperformed Random Forest, capturing up to 59% of actual positive cases in Feature Group 1. This highlights its ability to identify positive cases more effectively. In contrast, Random Forest exhibited lower recall, indicating that it missed a larger proportion of positive cases.

The F1-score remained consistently low (~0.02) across all models, reflecting the imbalance between precision and recall. This metric underscored the challenge of achieving a balanced performance given the dataset's characteristics.

Logistic Regression also outperformed Random Forest in AUC-ROC scores across all feature groups. Its highest score, 0.683, was achieved using Feature Group 1, demonstrating its superior ability to distinguish between positive and negative cases within the dataset.

Hyperparameter Fine-Tuning

The objective of the hyperparameter fine-tuning process was to enhance the performance of the best-performing Random Forest model from Group 1.

A grid search was employed to explore a range of hyperparameter values, as shown in the table below. The hyperparameters evaluated during the tuning process included the number of estimators (`n_estimators`), the maximum depth of the trees (`max_depth`), the minimum number of samples required to split an internal node (`min_samples_split`), the minimum number of samples required to be at a leaf node (`min_samples_leaf`), and the maximum number of features considered for splitting a node (`max_features`).

Parameter	Values Explored
`n_estimators`	[100, 200]
`max_depth`	[10, 20]
`min_samples_split`	[2, 5]
`min_samples_leaf`	[1, 2]
`max_features`	['sqrt']

Table 3. Hyperparameter values explored during the grid search for fine-tuning the Random Forest model in Feature Group 1.

Parameter	Best Value
`n_estimators`	100
`max_depth`	10
`min_samples_split`	5
`min_samples_leaf`	1
`max_features`	'sqrt'

Table 4. Optimal hyperparameter values identified for the Random Forest model in Feature Group 1 using 5-fold cross-validation.

Following the grid search with 5-fold cross-validation, the optimal hyperparameter values were then identified. The optimized Random Forest model achieved an AUC-ROC score of 0.652, marking a 7.6% improvement compared to the untuned model.

Feature Importance Analysis

The primary objective of the feature importance analysis was to assess the contribution of individual features to the overall performance of the model. Two distinct methods were employed to evaluate feature importance: impurity-based importance and permutation-based importance. Impurity-Based Importance measures the average decrease in impurity across all trees in the Random Forest model. This metric provides insight into how effective each feature is in splitting the data and reducing uncertainty within the model. Permutation-Based Importance involves evaluating the effect of randomly shuffling a feature's values on the model’s performance. This method helps quantify the impact of each feature on the predictive power of the model, with greater performance degradation indicating higher feature importance.

Findings:

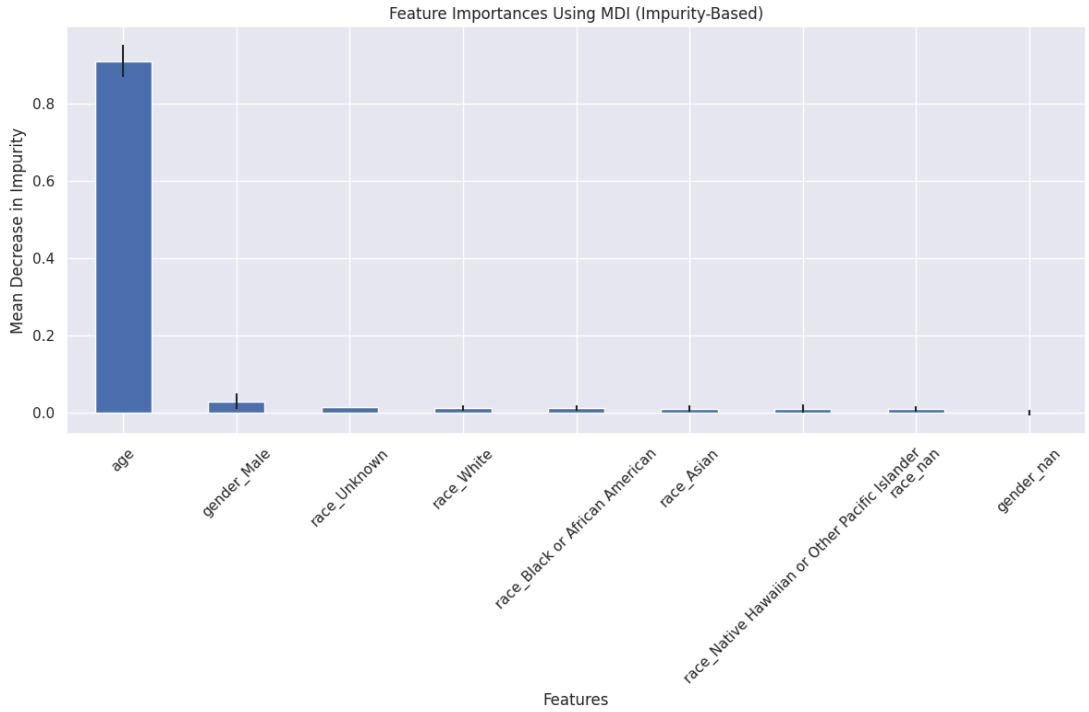

Figure 2. Impurity-based feature importance for Feature Group 1 Demographics using the Random Forest model, illustrating the mean decrease in impurity across all decision trees.

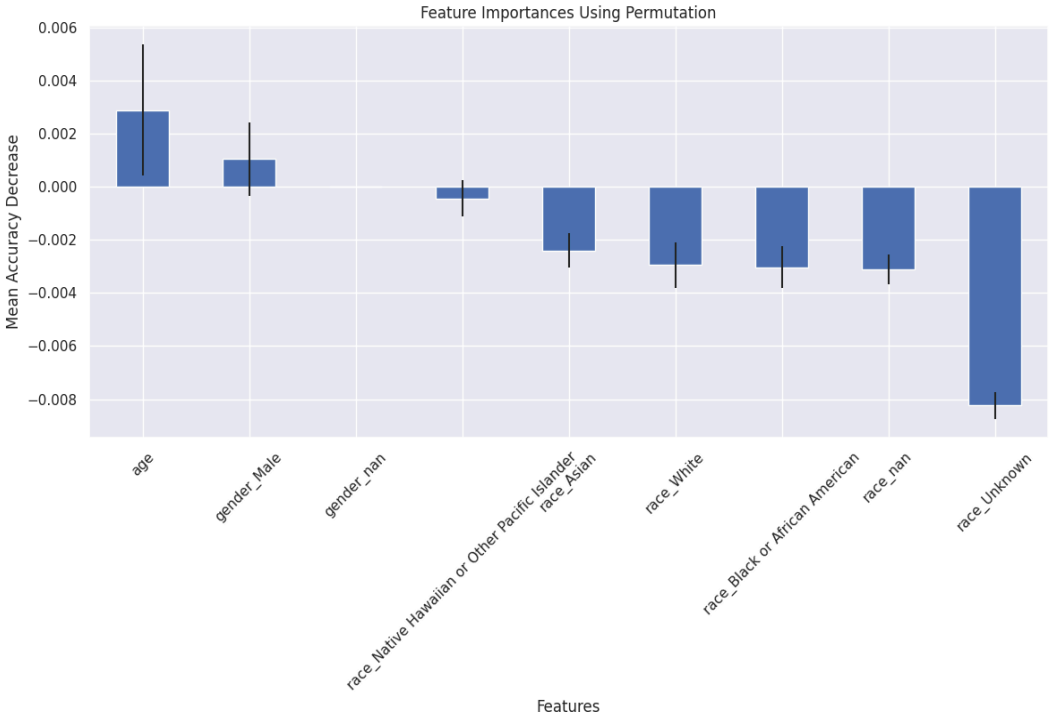

Figure 3. Permutation-based feature importance for Feature Group 2 (Demographics + Conditions) using the Random Forest model, showing the impact of shuffling feature values on model performance.

The results of the analysis revealed that, in both approaches, age was the most significant feature in the model. Under impurity-based importance, age contributed 90.9% of the total impurity reduction, suggesting that age-based splits were highly effective at distinguishing between positive and negative cases. The permutation-based analysis confirmed this finding, with age showing the most substantial positive impact on the model's performance when evaluated through feature shuffling.

On the other hand, the gender and race features (e.g., Gender_Male, Race_Unknown, Race_White) exhibited relatively low or negligible importance across both methods. Specifically, the permutation-based analysis revealed that these features had minimal influence on the model's performance. Interestingly, certain race-related features, such as Race_Unknown and Race_White, exhibited negative permutation importance, implying that the model performed slightly better when these features were shuffled. This could indicate potential issues such as multicollinearity or possible overfitting to noise, suggesting that these features may not provide meaningful information to the model.

Conclusions

Our analysis revealed several key findings. We observed that Logistic Regression consistently outperformed other models across all feature groups in identifying positive cases (recall). It emerged as the best-performing model, achieving an AUC-ROC of 0.683 using demographic features alone, suggesting it was well-suited to the characteristics of our dataset. In contrast, Random Forest struggled with recall, particularly when handling more complex feature sets. Its lower AUC-ROC values indicated that it was less effective than Logistic Regression in distinguishing between classes. Adding condition and drug-related features (Feature Groups 2 and 3) did not significantly improve model performance. We noted that Logistic Regression performed best with demographic features alone, indicating that the additional features offered limited predictive value. Furthermore, the significant class imbalance in our dataset heavily influenced model performance.

Future Improvements

In future work, we aim to address class imbalance more effectively by implementing resampling techniques such as Synthetic Minority Over-sampling Technique (SMOTE)

or undersampling. Additionally, we plan to explore advanced feature engineering approaches, including incorporating features like time intervals between visits, patterns of chronic disease progression, and detailed medical history. To further improve performance, we will experiment with alternative models such as XGBoost or LightGBM, which are known to perform well on imbalanced datasets.

Individual Contributions

- **Austin Cherian** - Extracted the age feature, ran the logistic regression and random forest models, did the hyperparameter tuning, and the feature importance analysis. Additionally, I worked on the presentation slides and final report.
- **Hanane Bousfoul -** Calculated the outcome variable, identified and grouped the conditions of interest, and transformed them into binary features. Additionally, I merged demographic data, analyzed the results from various models, and worked on the presentation slides and final report.
- **Natalie Ellis -** Performed EDA on observations and identified high-risk admissions using keyword searches. I also analyzed the medications table, categorizing high risk drugs into relevant groups and then created binary flags for these categories. I also ran the models for Feature Group 2 and worked on the presentation slides and final report.

References

1. Bionetworks S. EHR DREAM Challenge - Patient Mortality Prediction. Synapse.org. Published 2024. Accessed December 21, 2024. https://www.synapse.org/Synapse:syn18405991/wiki/589657
2. Charlson Comorbidity Index (CCI). MDCalc. Published 2024. Accessed December 21, 2024. https://www.mdcalc.com/calc/3917/charlson-comorbidity-index-cci