Myocardial Infarction Data Analysis

Amy Wang

December 16, 2024

1 Introduction

Myocardial infarction (heart attack) is caused by the obstruction of blood flow to the heart muscle, leading to ischemia and necrosis. This project focuses on analyzing data related to myocardial infarction (MI), a critical condition that remains a leading cause of death worldwide. The dataset we used includes 1,700 patient records with 124 features, capturing a wide range of clinical, diagnostic, and biochemical information.

The primary goal of our study is to reduce the dimensionality of the data by selecting the most relevant features using both parametric and non-parametric approaches. The second objective is to predict the target variable, "let_is," based on the selected features and compare the performance of parametric methods, such as logistic regression, and non-parametric methods, such as decision trees. By exploring these techniques, we aim to better understand the factors that influence MI outcomes and provide insights that could improve clinical decision-making and patient care.

2 Data Preprocessing

This study's dataset consists of 1,700 records and 124 features. In terms of feature types, after integration it was found that the dataset contains 110 floating-point features (float64) and 14 integer features (int64). The number and types of features indicate that the data source is relatively rich and diverse, encompassing medical information from multiple dimensions.

2.1 Data Analysis

2.1.1 Response Variable: let_is

The target variable in the dataset is a categorical variable called *let_is*, which includes 8 categories (0, 1, 2, 3, 4, 5, 6, 7). However, the class distribution is highly imbalanced: category 0 accounts for approximately 84.1% of the data and is thus the predominant class, while all other categories have proportions below 10%. In particular, categories 5 and 2 each account for less than 1%, implying that during subsequent modeling, one must consider strategies such as re-sampling, weighting, or other balancing approaches to improve model performance on minority classes and prevent excessive bias toward predicting the majority class.

2.1.2 Missing Values

The dataset exhibits pervasive missing values, with uneven distribution of these missing rates. Some features (such as *s_ad_kbrig* and *d_ad_kbrig*) have over 1,000 missing values, meaning a missing rate of more than 50%, which poses challenges for effective feature utilization. Additionally, certain features (like *ibs_nasl* and *kfk_blood*) have missing rates close to 95%, nearly fully missing, which raises questions about their utility in analysis and modeling. Possible approaches include dropping these features, substituting them with derived variables, or employing specific imputation strategies. Conversely, some features have no missing values at all, such as *id*, *sex*, and *let_is*. These features can provide stable indices, grouping parameters, or labels for subsequent analyses and modeling.

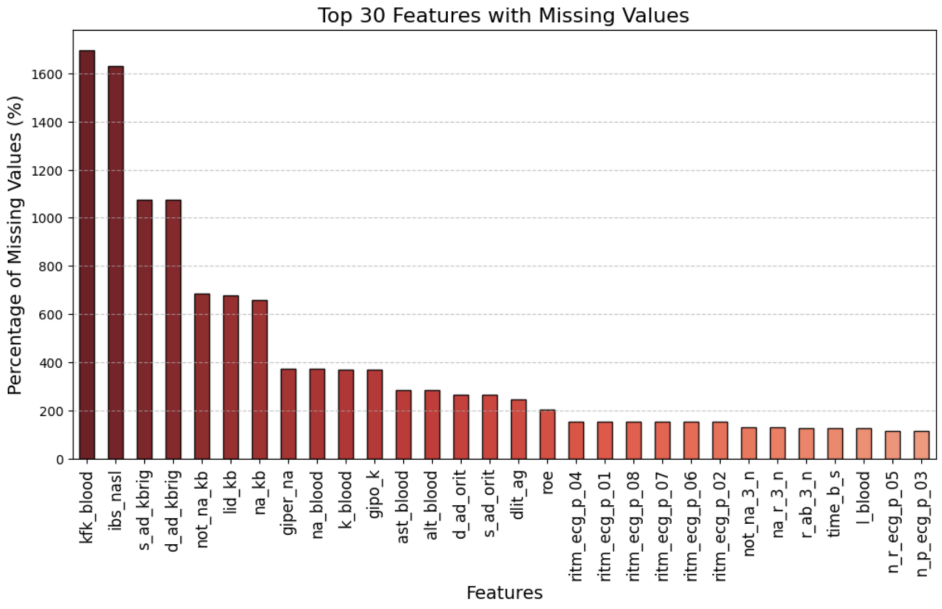

FIGURE 1: Data Missing

2.1.3 Feature Names and Meanings

Most feature names consist of English abbreviations or medical terms closely related to clinical and diagnostic information. For example, basic demographic features like *age* (indicating the patient's age) and *sex* (indicating the patient's gender) can provide fundamental stratification information for subsequent model building. Features related to medical history and symptoms—such as *inf_anam* (indicating medical history) and *sim_gipert* (indicating hypertension symptoms)—help models capture pathological characteristics and risk factors. The dataset also includes an array of biochemical indicators (such as *k_blood* for blood potassium concentration and *na_blood* for blood sodium concentration), which can help assess a patient's physiological state. Finally, diagnostic result features (like *fibr_preds*, *zsn*, etc.) may represent coded forms of clinical judgments, tests, or monitoring outcomes, providing valuable clinical references for model prediction and interpretation.

2.1.4 Summary

Overall, the feature dimensions, distributions, and missing data conditions in this dataset indicate the need for thorough data cleaning and preprocessing before further analysis and modeling. This includes addressing class imbalance and handling missing values. Additionally, a deep understanding of feature meanings and characteristics will assist in effective feature engineering, thereby optimizing the model's predictive performance and medical interpretability.

2.2 Missing inputation

In the data preprocessing workflow, the first step involves removing rows with a high proportion of missing values. Specifically, any record for which more than 20% of the features are missing will be deleted. Next, features are categorized based on their missing rate into three groups: low missing rate (less than 5%), moderate missing rate (5% to 30%), and high missing rate (above 30%). Features with a high rate of missing values should be considered for removal altogether. Afterward, features are split into continuous and categorical types, treating those with a float64 type as continuous variables, and all others (such as int64, category, object, etc.) as categorical variables. The imputation strategy is then chosen according to the variable type: for continuous features, the median is used to fill in missing values, while for categorical features, the mode is applied. Implementing these measures effectively mitigates the adverse impact of missing data on model performance, ensuring stability and reliability in subsequent analysis and modeling. Specifically:

1. **Drop Rows with High Missing Ratios**

- **Criteria:** Drop rows where more than 20% of the values are missing.

2. **Categorize Features by Missing Rate**
3. Features are divided into three groups based on their missing data ratio:
   (a) **Low Missing:** Missing rate $< 5\%$.
   (b) **Medium Missing:** Missing rate between 5% and 30%.
   (c) **High Missing:** Missing rate $> 30\%$.
4. **Action for High Missing Features:** Drop these columns if they exist in the dataset.
5. **Separate Continuous and Categorical Variables**
   - Continuous variables are identified as those of type `float64`.
   - Categorical variables include all non-`float64` data types.
6. **Impute Missing Values**
   - **For Continuous Variables:**
     - Use **median imputation** to replace missing values.
   - **For Categorical Variables:**
     - Use **mode (most frequent value) imputation** to replace missing values.
   - This ensures the imputed values are robust and suitable for their respective data types.

2.3 Dimension Reduction

In our dataset under study, the initial set comprised 115 features. High-dimensional data of this nature often comes with issues of multicollinearity. Because certain features exhibit strong linear relationships with one another, modeling with all features directly not only complicates the interpretability of the model, but also reduces its robustness. In such cases, a simple correlation heatmap is insufficient to effectively capture the complex interrelationships among so many features, and the actual contribution and statistical significance of some features may be overshadowed. Under these circumstances, employing appropriate dimensionality reduction techniques to eliminate redundant features and lessen inter-feature interference is highly beneficial for subsequent analysis and modeling.

How to Reduce Dimensionality

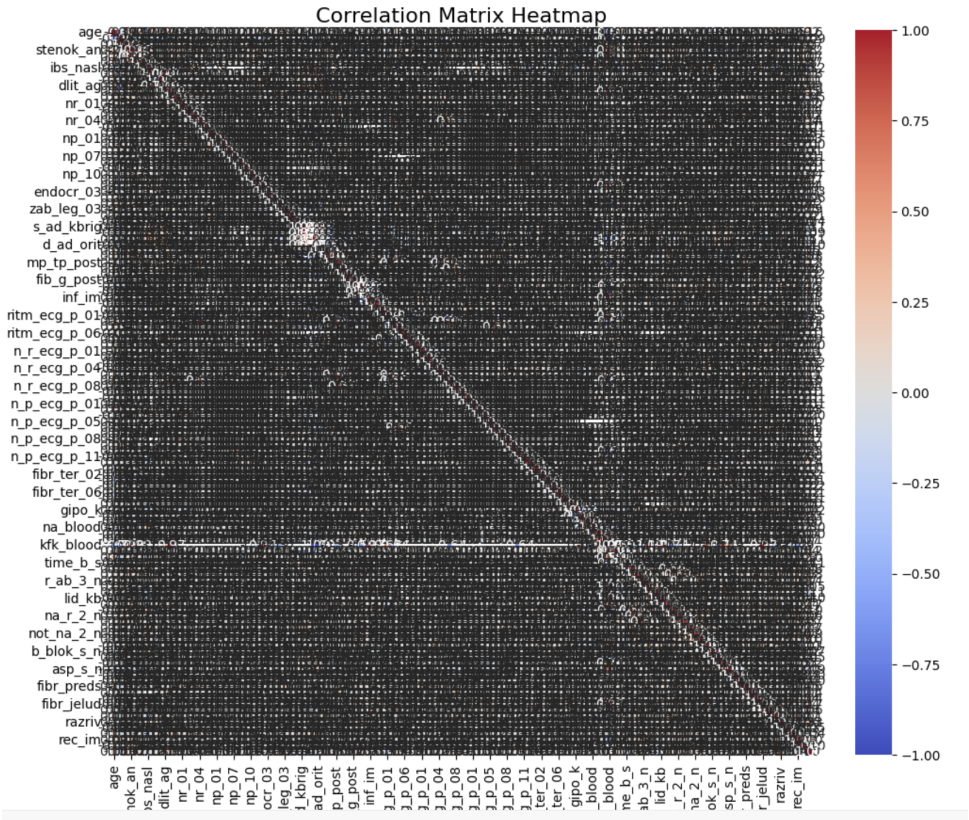

FIGURE 2: Correlation Matrix

1. **Calculate Variance Inflation Factor (VIF):** Compute the VIF for each continuous feature to quantitatively assess the degree of linear correlation between that feature and the others.

2. **Set a Removal Criterion:** Based on empirical guidance or industry standards, establish a threshold (e.g., VIF $\leq$ 5). Any feature whose VIF value exceeds this threshold is considered to exhibit high multicollinearity and can be flagged for removal.

3. **Result**

   - **Before Filtering:**
     - Total Columns: 115
   - **After Filtering:**
     - Remaining Columns: 100
   - **Reduction:**
     - Removed 15 Columns with high multicollinearity.

3 Non-Parametric Methods

We used some non-parametric methods, such as Random Forests and Classification Trees which are able to offer powerful tools for analyzing complex datasets without making strong assumptions about the underlying data distribution. These methods are particularly suited for datasets with non-linear relationships and interactions between predictors.

3.1 Feature Selection using Random Forests

Firstly is Random Forests. This is an ensemble learning technique, and we employed it to identify the most important predictors for the target variable, *let_is*. This method help build multiple decision trees and aggregates their predictions to reduce variance and improve robustness.

Feature Importance Feature importance in Random Forests is derived from the reduction in impurity achieved by splits involving a specific predictor. For a feature j, its importance is calculated as:

$$I(j) = \sum_{t=1}^{T} \sum_{s \in S_j} \left(i_s - \frac{n_{s_L}}{n_s} i_{s_L} - \frac{n_{s_R}}{n_s} i_{s_R} \right),$$

where T is the number of trees, S_j is the set of splits involving j, and i_s represents node impurity.

Feature Selection The top 20% of features were selected based on importance scores, resulting in key predictors such as *razriv*, *k_sh_post*, *roe*, and *ast_blood*, among others.

3.2 Classification Tree with Hyperparameter Tuning

Classification Trees were used to classify the target variable based on the selected features. These trees recursively partition the data by maximizing the reduction in impurity at each split.

Impurity Measures For a given split, the information gain is defined as:

$$IG = i_p - \frac{n_L}{n_p} i_L - \frac{n_R}{n_p} i_R,$$

where i_p, i_L, and i_R are the impurities of the parent, left, and right nodes, and n_L, n_R, and n_p are the corresponding sample counts.

Hyperparameter Tuning GridSearchCV was used to optimize tree depth, minimum samples per split, and minimum samples per leaf. The optimal parameters were:

$$\text{max\_depth} = 3, \quad \text{min\_samples\_split} = 2, \quad \text{min\_samples\_leaf} = 2.$$

3.3 Evaluation and Results

The final decision tree model, trained on the top 20% of features, achieved a test accuracy of 89.81%. This demonstrates the effectiveness of non-parametric methods in capturing complex relationships in the dataset.

4 Parametric Methods: Multinomial Logistic Regression

In the previous section, we explored non-parametric methods such as random forests and classification trees, which are powerful tools for handling complex, non-linear relationships in the data without making strong assumptions about its underlying structure. While these methods excel in flexibility and predictive power, they often lack interpretability, making it challenging to understand the precise contributions of individual predictors.

To complement these approaches, this section focuses on a parametric method, specifically multinomial logistic regression, which provides a more interpretable framework for multi-class classification tasks. By leveraging feature selection techniques such as Recursive Feature Elimination (RFE), SelectKBest with test accuracy, and SelectKBest with stratified cross-validation, we aim to identify the most relevant predictors and build an efficient, interpretable model to predict complications and causes of death after myocardial infarction.

Multinomial logistic regression is an extension of binary logistic regression used when the dependent variable has more than two classes. In this case, we have eight different classes in dependent variable "let_is".

For a multi-class target variable Y with K classes, the model estimates the probabilities of each class given a set of predictors X.

The relationship is modeled as:

$$P(Y = k|X) = \frac{\exp(\beta_{k0} + \beta_{k1}X_1 + \cdots + \beta_{kp}X_p)}{\sum_{j=1}^{K} \exp(\beta_{j0} + \beta_{j1}X_1 + \cdots + \beta_{jp}X_p)} \quad \text{for } k = 1, \ldots, K$$

where: - $P(Y = k|X)$ is the probability of class k given predictors X, - β_{kj} are the regression coefficients estimated via maximum likelihood.

The "lbfgs" solver in python was used to handle multi-class problems efficiently.

After having the classification model, next we are going to discuss three different feature selection methods.

4.1 Feature Selection

4.1.1 SelectKBest method with Cross-Validation

SelectKBest is a univariate feature selection method that selects the top **k** predictors based on their statistical relationship with the target variable.

"f_classif" is used as the scoring function, which applies an ANOVA F-test to evaluate the association between each predictor and the target variable.

Next, **Cross-Validation (CV)** is used to validate the performance of the model built using the selected features.

All the relavent codes are in Appendix section. Here is the output:

```
Optimal Number of Features: 1
Selected Features: ['k_sh_post']
Max Test Accuracy: 0.8726
```

TABLE 1: SelectKBest method with Cross-Validation Results

- The results show that the selection of only one feature provided the best performance (highest accuracy) during cross-validation. The selected feature is "**k_sh_post**", and the logistic regression model trained on this single feature achieved 87.26% accuracy when evaluated on the test set.
- "k_sh_post" has a very strong statistical association with the target variable "let_is", as it alone outperformed other feature combinations in cross-validation. This suggests that it might be a key predictor of complications or causes of death in the data set.
- However, it is intuitive to see that selecting only one predictor can be problematic: although having a single predictor is extremely straightforward and easy to interpret, relying on a single feature may overlook interactions with other predictors or fail to capture the full complexity of the data.

4.1.2 SelectKBest method with 5-fold Stratified Cross-Validation

From the previous part, we observed that simply using Cross-Validation can lead to certain issues. We address the following two key concerns:

Ratio of Observations to Features: The ratio between the number of observations and features is not large enough, meaning we have too many predictors relative to the sample size. Using 10-fold Cross-Validation in such cases may result in folds with very few observations, leading to unreliable and problematic results. To mitigate this, we now apply 5-fold Cross-Validation, which ensures each fold contains more observations, improving stability and reliability.

Class Imbalance in Folds: When splitting the data into folds, there is a risk that some folds might not contain all classes of the response variable, leading to unsatisfactory model performance. To address this, we introduce **Stratified Cross-Validation**. Stratified K-Fold ensures that each fold maintains the same class distribution as the full dataset. This is particularly important for multi-class problems like let_is, where class imbalances may exist.

Thus in this part we applied SelectKBest feature selection method with 5-fold Stratified Cross-Validation to improve the robustness and reliability of our multinomial logistic regression results.

The result shows that:

```
Optimal Number of Features: 4
Selected Features: ['zsn_a', 'nr_01', 'k_sh_post', 'fibr_jelud']
Max Cross-Validation Accuracy: 0.8586
Test Accuracy: 0.8662
```

TABLE 2: SelectKBest method with 5-fold Stratified Cross-Validation

- After evaluating different subsets of features, the best performance was achieved using 4 features: **"zsn_a", "nr_01", "k_sh_post", "fibr_jelud"**. And we can see that "k_sh_post" is again one of the most important feautre, which reinforced its predictive power.
- When the logistic regression model was trained on the full training set using the 4 selected features and evaluated on the test dataset, it achieved an accuracy of 86.62%. The close alignment between the cross-validation accuracy (85.86%) and test accuracy (86.62%) indicates that the model generalizes well to unseen data.
- However, when comparing this result with that of the previous method, the addition

of 3 more features led to a slight decrease in test accuracy. This means that there might be some trade of between model simplicity and performance stability.

From the results of the two methods using the SelectKBest framework with Cross-Validation, we observe that the selected feature subsets are too small to be entirely satisfying. In particular, selecting only one or four features out of over one hundred predictors seems unrealistic, as it oversimplifies the complexity of heart disease causes. Myocardial infarction and its complications are inherently multifactorial, likely involving the interplay of numerous predictors rather than being determined by a handful of features.

To address this limitation, we introduce a more comprehensive feature selection method: Recursive Feature Elimination (RFE). Unlike univariate approaches such as SelectKBest, RFE evaluates predictors in the context of a model, recursively eliminating the least important features based on their contributions to the logistic regression model. This method considers potential interactions between predictors, enabling us to identify a more meaningful subset of features while balancing interpretability and performance.

4.1.3 Recursive Feature Elimination (RFE)

RFE is a feature selection technique used to determine the most relevant features for a model. It recursively removes the least important features by training the model iteratively.

Steps Involved:

- Standardize predictors using StandardScaler to ensure equal contribution of all features.
- Train a multinomial logistic regression model iteratively with different numbers of features (from 1 to all predictors).
- Evaluate model performance using test accuracy.
- Identify the subset of predictors that yields the maximum test accuracy.

And here we have the results:

```
Optimal Number of Features: 45
Selected Features: ['sex', 'inf_anam', 'stenok_an', 'ibs_post', 'sim_gipert', 'zsn_a', 'nr_11', 'nr_03', 'nr_04', 'endocr_01', 'endocr
_02', 'zab_leg_01', 'zab_leg_02', 'zab_leg_03', 'k_sh_post', 'mp_tp_post', 'ant_im', 'lat_im', 'inf_im', 'post_im', 'im_pg_p', 'n_r_ec
g_p_03', 'n_r_ecg_p_06', 'n_p_ecg_p_12', 'gipo_k', 'alt_blood', 'ast_blood', 'roe', 'time_b_s', 'r_ab_1_n', 'r_ab_2_n', 'r_ab_3_n', 'n
itr_s', 'not_na_1_n', 'not_na_2_n', 'b_blok_s_n', 'ant_ca_s_n', 'gepar_s_n', 'preds_tah', 'jelud_tah', 'fibr_jelud', 'otek_lanc', 'raz
riv', 'zsn', 'p_im_sten']
Max Test Accuracy: 0.9044585987261147
```

Table 3: Recursive Feature Elimination (RFE)

- There are 45 important features selected by RFE, with the max test accuracy to be

90.4%. This likely due to RFE's ability to consider feature interactions in the context of the logistic regression model.

- The improvement in accuracy indicates that RFE is suitable for large and more complexed dataset, for instance, reflecting the multifactorial nature of heart disease.
- However, one important point to discuss is that RFE required nearly forty additional features to achieve an small improvement in accuracy. What underlying issue might this reveal? Also, since we select 45 predictors, is there a risk of over-fitting?
- Additionally, this method is very time consuming.
- For further study, we might want to explore other alternative methods, such as regularized regression (LASSO or Ridge), which can automatically penalize the inclusion of too many predictors and mitigate over-fitting.

5 Mixed Method: Random Forest + Recursive Feature Elimination (RFE)

After discussing non-parametric and parametric methods separately, we became curious: what if we combine these two approaches? Could this combination yield even better results? Hence in this section, a two-step feature selection process is applied to improve the performance of the multinomial logistic regression model. The workflow combines the strengths of Random Forest (a non-parametric method) and Recursive Feature Elimination (RFE) to identify the most relevant predictors.

Steps Involved:

- Feature Ranking with Random Forest: Train a Random Forest model to calculate feature importance. Select the top 20% of features based on their importance scores.
- RFE with Logistic Regression: Perform RFE to iteratively select subsets of features from the top-ranked features. Use Stratified K-Fold Cross-Validation (5-fold i this case) to evaluate model accuracy for different subset sizes. Identify the subset with the highest cross-validation accuracy as optimal.
- Final Model Evaluation: Train the logistic regression model on the optimal feature subset. Then evaluate the model's performance on the test set.

And here are the results:

```
Step 1 - Features Selected by Random Forest: ['razriv', 'k_sh_post', 'roe', 'ast_blood', 'alt_blood', 'time_b_s', 'stenok_an', 'zsn_a',
'na_r_1_n', 'lat_im', 'nitr_s', 'ant_im', 'inf_im', 'inf_anam', 'ibs_post', 'r_ab_1_n', 'otek_lanc', 'not_na_1_n', 'ant_ca_s_n', 'n_p_ec
g_p_12']
Optimal Number of Features (Step 2): 11
Selected Features (Step 2): ['razriv', 'k_sh_post', 'zsn_a', 'na_r_1_n', 'lat_im', 'nitr_s', 'ant_im', 'inf_im', 'ibs_post', 'r_ab_1_n',
'otek_lanc']
Max Cross-Validation Accuracy: 0.8946
Test Accuracy: 0.8917
```

TABLE 4: Mixed Method: Random Forest + Recursive Feature Elimination (RFE)

- After Random Forest and RFE, the optimal subset size was determined to be 11 features: **razriv, k_sh_post, zsn_a, na_r_1_n, lat_im, nitr_s, ant_im, inf_im, ibs_post, r_ab_1_n, otek_lanc**. Here, we see **k_sh_post** appears in results for all methods we discussed in this project, indicating that it truly is one of the most important feature.
- The 5-flod cross-validation accuracy (89.46%) and test accuracy (89.17%) are closely aligned, indicating good model generalizability and low risk of overfitting.
- When comparing to the previous methods, this mixed method achieves a strong balance between model performance, feature reduction, and interpretability.

6 Conclusion and Discussion

Handling missing values and addressing the correlation between predictors are critical steps when working with new data. Careful and appropriate data processing procedures lay a solid foundation for meaningful and reliable data analysis.

In this project, we conducted data analysis from three perspectives: parametric methods, non-parametric methods, and mixed methods:

Method	**Optimal Features**	**Cross-Validation Accuracy**	**Test Accuracy**	**Key Strength**
SelectKBest (10-fold CV)	1	N/A	**87.26%**	Simplicity (1 feature, fast selection).
SelectKBest (Stratified 5-fold CV)	4	85.86%	**86.62%**	Robust CV evaluation, smaller set.
RFE	45	N/A	**90.45%**	Captures feature interactions well.
Random Forest + RFE	11	89.46%	**89.17%**	Balances non-linear and linear models.

TABLE 1: Comparison of Feature Selection Methods Applied to Multinomial Logistic Regression

After reviewing all the outcomes, we conclude that the method combining Random Forest with RFE performs the best. While achieving a relatively high accuracy (only slightly lower than that of standalone RFE), it selects a reasonable number of features, allowing for better interpretation and model simplicity.

While non-parametric methods offer flexibility and strong predictive performance, they are computationally intensive and may be prone to overfitting. Additionally, their interpretability is limited compared to parametric models like multinomial logistic regression.

For future studies, we believe that better handling of missing values will be the most valuable topic to explore.

References

[1] GeeksforGeeks. *Feature Importance with Random Forests*. Accessed: December 15, 2024. URL: https://www.geeksforgeeks.org/feature-importance-with-random-forests/.

[2] Niranjan Ojha and Amit S. Dhamoon. "Myocardial Infarction". In: *StatPearls* (Aug. 2023). Last Update: August 8, 2023. URL: https://www.ncbi.nlm.nih.gov/books/NBK537076/.