Methodology Report: Logistic Regression-Based Feature Selection and Classification

Aayush Adhikari

May 31, 2025

1 Introduction

This report details the methodology implemented to perform supervised binary classification on a high-dimensional biomedical dataset. The task involved selecting informative features from over 3200 anonymized features and predicting class probabilities using Logistic Regression. Herein, we discuss the preprocessing strategy, the feature selection approach (RFECV), key hyperparameters, validation methodology, and the rationale behind each step.

2 Data Preprocessing and Feature Engineering

Initially, the dataset contained 3215 features after basic filtering. However, several issues were identified, including a high proportion of missing values, highly correlated features, and infinite or non-finite values.

2.1 Missing Value Handling

Features exhibiting more than 30% missing values were discarded to maintain data integrity and reduce the risk of imputation-induced bias. Post-removal, we had a dataset with manageable missingness, which was subsequently imputed using the median value for robustness against outliers and skewed distributions.

2.2 Correlation-Based Feature Filtering

Due to the high dimensionality, multicollinearity posed significant risks of inflating variance and causing instability in feature selection. Therefore, we applied a correlation filter to remove highly correlated features (*threshold* > 0.95):

```
def remove_highly_correlated(df, threshold=0.95):
    corr_matrix = df.corr().abs()
    upper = corr_matrix.where(np.triu(np.ones(corr_matrix.shape), k=1).astype(bool)
    to_drop = [column for column in upper.columns if any(upper[column] > threshold)
    return df.drop(to_drop, axis=1)
```

This effectively reduced redundant information and minimized computational overhead in subsequent feature selection steps.

2.3 Handling Infinite Values

Infinite (`inf`) and negative infinite (`-inf`) values were identified and replaced by `NaN` to allow seamless integration with the median imputation strategy. This ensured numerical stability across modeling steps:

```
X_reduced.replace([np.inf, -np.inf], np.nan, inplace=True)
```

After these steps, the dataset was reduced from 3215 to 822 features, with minimal missingness (only 2 values remained).

3 Feature Selection Strategy

Recursive Feature Elimination with Cross-Validation (RFECV) was implemented to rigorously select the most predictive features. RFECV operates by iteratively eliminating the least important features based on model coefficients (Logistic Regression), evaluating performance at each step via cross-validation, and ultimately selecting an optimal feature subset.

3.1 Implementation Details

RFECV was configured with the following parameters:

- `estimator`: Logistic Regression (L1 penalty).
- `step`: 10 (number of features removed per iteration).
- `cv`: 5-fold stratified cross-validation.
- `scoring`: Area Under ROC Curve (ROC-AUC).
- `min_features_to_select`: 20, ensuring a robust minimum feature count.
- `n_jobs`: -1 (parallelization for efficiency).

The cross-validation strategy specifically used stratified folds to maintain class distribution consistency, critical given class imbalance.

4 Selected Feature Subset

After applying RFECV, the final feature subset comprised 152 features, significantly reducing the original dimensionality:

Example subset (partial): Feature_13, Feature_18, Feature_27, Feature_61, Feature_65, Feature_72, Feature_96, Feature_116, etc.

The complete set of selected features was retained for subsequent predictive modeling and testing.

5 Model Architecture and Key Hyperparameters

The final predictive modeling step employed Logistic Regression, recognized for its interpretability, efficiency, and effective handling of sparse feature sets post-L1 regularization. The pipeline included:

1. **Median Imputation**: Robust against outliers.
2. **Standard Scaling**: Critical for Logistic Regression convergence and performance.
3. **Logistic Regression (L1 regularized)**: To induce sparsity and prevent overfitting, hyperparameters were explicitly set:
   - `solver='liblinear'`: Efficient for L1 penalty.
   - `class_weight='balanced'`: Adjust for class imbalance.
   - `max_iter=1000`: Ensure convergence.

The hyperparameter choices were explicitly justified by their necessity to control overfitting and computational stability.

6 Cross-Validation Scheme

A comprehensive validation framework was essential to objectively evaluate predictive performance and avoid data leakage:

- **Feature Selection Validation**: Implemented using RFECV with a 5-fold stratified cross-validation scheme to select features that generalize well beyond the training set.
- **Final Model Validation**: An independent train-validation split (80/20 stratified) assessed generalization and ensured unbiased estimates of predictive performance metrics.

Stratified splits ensured that each fold and validation subset retained the class distribution, a critical step to accurately evaluate metrics sensitive to imbalance, such as sensitivity and specificity.

7 Model Evaluation Metrics

The following performance metrics were computed to thoroughly assess the model's predictive ability:

- **Accuracy**: Overall classification performance.
- **AUROC**: Ability to distinguish between classes across all thresholds.
- **Sensitivity (Recall)**: Proportion of actual positives correctly identified.

- **Specificity**: Proportion of actual negatives correctly identified.
- **F1-score**: Balance between precision and recall.

On the validation set, Logistic Regression achieved:

- Accuracy: 0.81
- ROC AUC: 0.88
- Sensitivity: 0.72
- Specificity: 0.87

These balanced performance metrics demonstrated the effectiveness of our feature selection and modeling strategy.

8 Reflections and Future Improvements

Strengths:

- Robust feature engineering reduced risk of overfitting.
- Clear, interpretable modeling pipeline suitable for biomedical applications.
- Rigorous validation strategy ensuring model reliability.

Limitations and Areas for Improvement:

- Limited interpretability due to anonymized features.
- Small sample size constraints.
- Potential improvements include ensemble methods, advanced hyperparameter tuning (e.g., Bayesian optimization), and exploration of non-linear models like XGBoost or Neural Networks with careful regularization.

9 Conclusion

Through detailed preprocessing, strategic feature selection via RFECV, and carefully tuned Logistic Regression, our methodology provided robust predictive performance on a challenging high-dimensional dataset. The outlined approach is reproducible and demonstrates key best practices for supervised classification tasks in high-dimensional settings.