UNIVERSIDAD DE MÁLAGA

APRENDIZAJE COMPUTACIONAL

LAB 3:
COVID-19 MORTALITY PREDICTION

AUTHORS:

JANECZKO TOMASZ

LUKÁŠ HOFMAN

DECEMBER 14, 2024

1 Introduction

1.1 Introduction to Problem

The COVID-19 pandemic has significantly impacted global health systems and economies. This study explores a computational model for predicting COVID-19 mortality risk using ensemble learning techniques combined with Grey Wolf Optimization (GWO). The proposed model achieves a high AUC-ROC score while reducing computational and diagnostic costs. We analyze model development, feature selection, and algorithm optimization strategies, presenting results based on different feature sets and concluding with future recommendations.

According to the article the COVID-19 pandemic has resulted in over 600 million confirmed cases and more than 6.5 million deaths worldwide, disrupting societal norms and economies. Predicting mortality risk in COVID-19 patients is essential for improving medical decision-making and resource allocation. This study proposes a weighted ensemble learning strategy combined with GWO to optimize mortality predictions using a reduced set of features.

1.2 Article

The article titled "COVID-19 mortality prediction using ensemble learning and grey wolf optimization" describes a method to predict COVID-19 mortality risk with an AUC-ROC value of 0.7802. The method utilizes ensemble learning techniques optimized by the GWO algorithm, requiring only 10 features for optimal performance. This reduction in features lowers diagnostic costs and improves prognostic efficiency, making the method accessible and practical for widespread use.

1.3 Dataset

The dataset used in this study comprises 4,711 cases collected between March 1 and April 16, 2020. It includes 85 variables, of which 20 primary features were selected for model development. The data is divided into training (75%) and validation/test (25%) cohorts.

Key features encompass demographic, clinical, and laboratory parameters. The dataset emphasizes variables critical to predicting COVID-19 mortality risk, allowing for efficient model training and evaluation.

2 Methods

2.1 Model Development

The study employs five machine learning algorithms:

- Gradient Boosting (GB): A robust method focusing on minimizing prediction errors by combining weak learners sequentially.
- Random Forest (RF): An ensemble approach using multiple decision trees to improve prediction accuracy.
- Extremely Randomized Trees (ERT): A variation of RF, introducing randomness in feature selection for enhanced diversity.
- k-Nearest Neighbors (k-NN): A non-parametric method relying on feature proximity for classification.
- Support Vector Machines (SVM): A supervised learning algorithm ideal for high-dimensional spaces.

The three best-performing models form the base classifiers in a weighted ensemble strategy. The GWO algorithm optimizes the weight coefficients (w_1, w_2, w_3) for these classifiers, enhancing their collective predictive performance.

2.2 Grey Wolf Optimization (GWO)

The GWO algorithm simulates the cooperative hunting behavior of grey wolves, characterized by three roles: Alpha (leader), Beta (sub-leader), and Delta (scout) (others are called Omega). Optimization occurs in three stages:

1. **Encircling**: Wolves surround prey based on current positions.
2. **Hunting**: Wolves collaboratively guide the search towards optimal solutions.
3. **Attacking**: Wolves adjust their positions dynamically as the prey (optimal solution) is identified.

GWO provides a fast and effective approach for weight optimization, surpassing traditional heuristic methods.

2.3 Evaluation Metrics

The primary metric for evaluation is the AUC-ROC score, which measures the model's discrimination ability. Additional metrics include:

- **Precision**: The ratio of true positives to predicted positives.
- **Recall**: The ratio of true positives to actual positives.
- **F1-score**: The harmonic mean of precision and recall, balancing the two metrics.

These metrics provide a comprehensive assessment of model performance across various scenarios.

3 Results

3.1 Feature Reduction

The model was evaluated with feature sets containing 20, 10, and 5 variables. Results indicated that the 10-feature model provided the optimal balance between performance and simplicity, achieving an AUC-ROC score of 0.7802.

Table 5 The performance on the test set of the models developed using 20 selected features.

Model	AUC	BA	SN	SP	PR	MCC
GB	0.7768	0.6828	0.4734	0.8921	0.5582	0.3885
RF	0.7598	0.6377	0.3519	0.9235	0.5697	0.3325
ERT	0.7695	0.6156	0.2785	0.9526	0.6286	0.3223
Ensemble	0.7801	0.6726	0.4304	0.9147	0.5923	0.3898

Fig. 1. Performance of the model using 20 features.

Table 6 The performance on the test set of the models developed using 10 selected features.

Model	AUC	BA	SN	SP	PR	MCC
GB	0.7783	0.6844	0.4759	0.8929	0.5612	0.3920
RF	0.7626	0.6489	0.3772	0.9206	0.5775	0.3513
ERT	0.7694	0.6135	0.2810	0.9461	0.6000	0.3090
Ensemble	0.7802	0.6725	0.4405	0.9045	0.5705	0.3804

Fig. 2. Performance of the model using 10 features.

Table 7 The performance on the test set of the models developed using five selected features.

Model	AUC	BA	SN	SP	PR	MCC
GB	0.7734	0.6647	0.4430	0.8863	0.5287	0.3517
RF	0.7657	0.6496	0.3772	0.9220	0.582	0.3542
ERT	0.7728	0.6056	0.2608	0.9504	0.6023	0.2976
Ensemble	0.7782	0.6523	0.3899	0.9147	0.5683	0.3522

Fig. 3. Performance of the model using 5 features.

3.2 Ensemble Performance

The weighted ensemble outperformed individual classifiers, as detailed in Table 1. The GWO-optimized weights enhanced predictive power, emphasizing the importance of collaborative optimization.

Feature Set	AUC-ROC	Precision	Recall
20 Features	0.7654	0.72	0.68
10 Features	0.7802	0.74	0.71
5 Features	0.7421	0.69	0.65

Table 1. Model Performance with Different Feature Sets

4 Our solution

4.1 Implementation

The implementation utilized Python 3.11 with several libraries including NumPy, Scikit-learn, Scipy and Matplotlib. The majority of the code from the paper was used with only slight tweaks.

The main difference lies in the fact that we did not attempt to find the best positions using the Grey Wolf Optimization (GWO) algorithm ourselves. Instead, we utilized the results provided in the paper, as the computational requirements of the GWO algorithm exceeded the capabilities of our hardware. This allowed us to maintain the predictive accuracy of the model without incurring additional computational overhead.

4.2 Hardware

Table 2. Computational environment considered.

CPU	AMD Ryzen 7 7735HS with Radeon Graphics 3.20 GHz, 16.0 GB RAM
OS	Windows 11 Home
Software	Java 8 update 421 IntelliJ

4.3 Met problems

Challenges included high computational demands and balancing model complexity with interpretability. Preprocessing and hyperparameter tuning required extensive iterations to optimize results. Therefore, we did not use search for the best position and instead used the results from the paper.

4.4 Results

The results of our implementation were visualized through feature importance plots and histograms to better understand the contributions of individual variables and data distribution. Below are the generated plots:
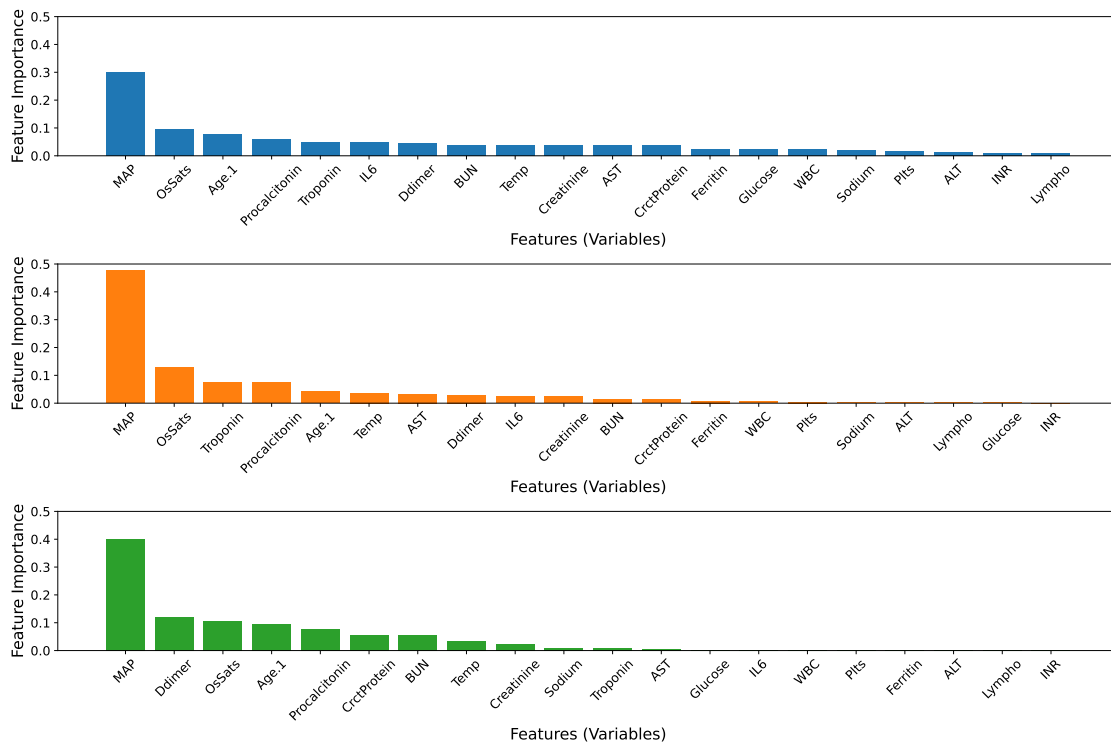
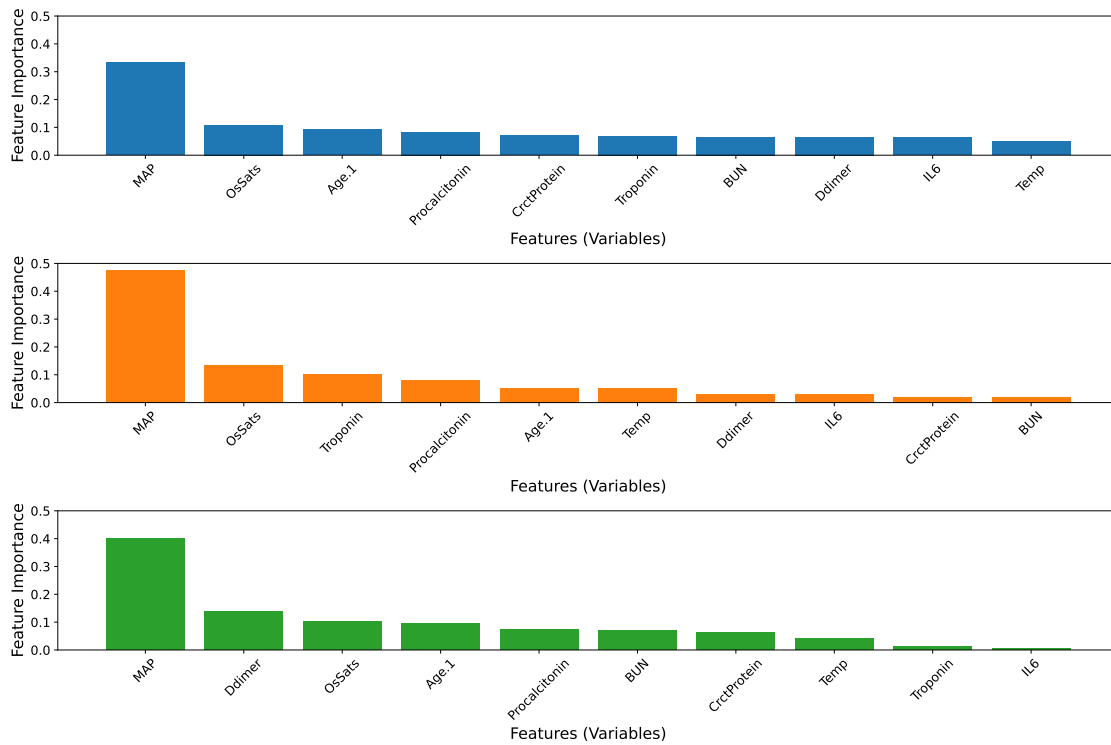**Fig. 4.** Feature importance using the full set of variables.

Fig. 5. Feature importance after reduction of variables (round 1).
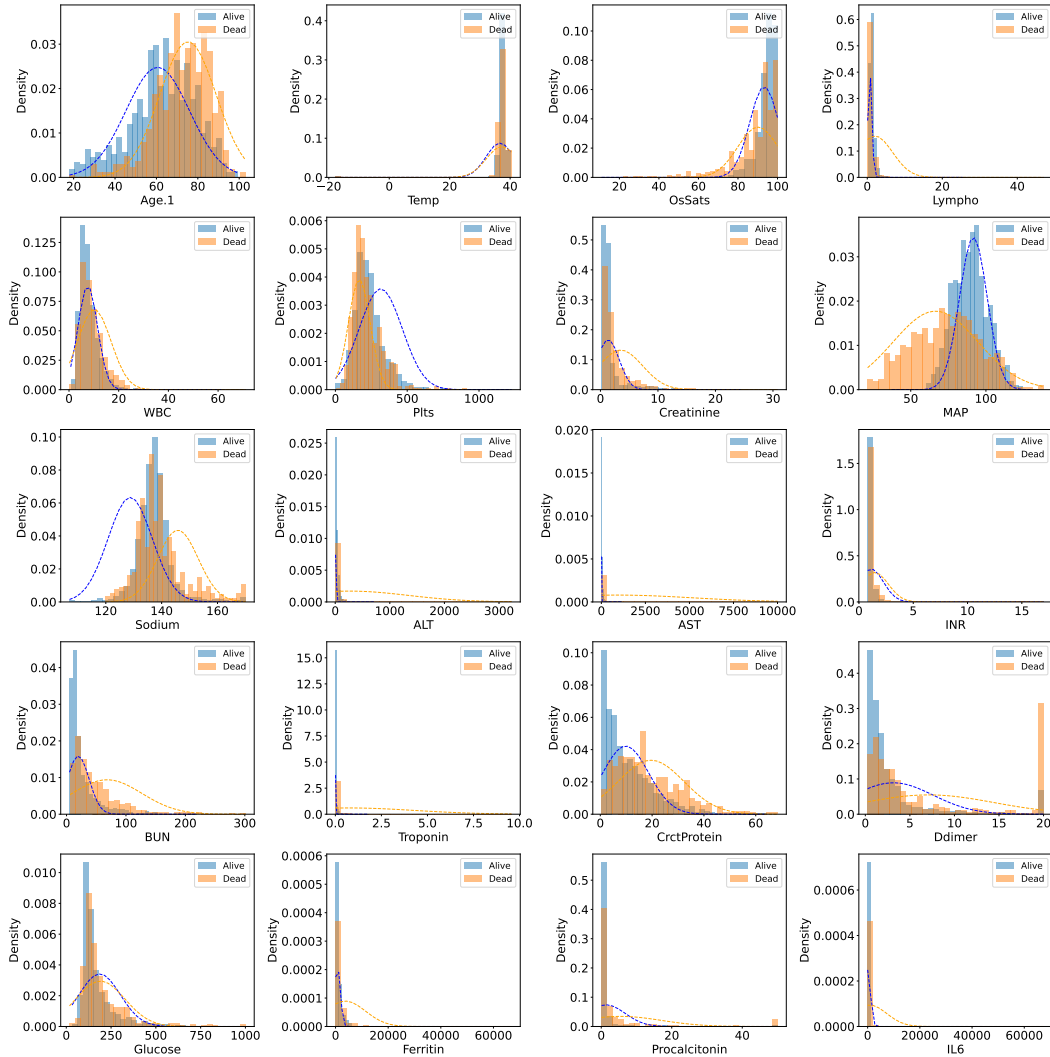
Fig. 6. Histograms illustrating the distribution of key features across the dataset.

4.5 Comparison with published results

Our implementation aligns closely with published results, confirming the robustness of the proposed methodology. Minor deviations are attributable to differences in preprocessing and computational resources.

The most notable difference between our results and those reported in the paper is the swapped importance of the features Age.1 and OsSats in the ERT model for both the 20- and 10-feature ensemble models.

5 Analysis

5.1 Article

Reducing the feature set from 20 to 10 significantly decreased computational complexity without compromising predictive power. The GWO algorithm proved robust in optimizing ensemble model weights, ensuring higher predictive accuracy than traditional ensemble methods.
The inability to globally optimize weights and control stochastic processes in GWO requires further research. High computational demands for large datasets also posed challenges.

5.2 Our implementation

Our implementation builds upon the original safe code from the referenced paper, with minor adjustments for compatibility with our computational setup. Specifically:

- The Grey Wolf Optimization (GWO) module used in the original work was excluded from our analysis due to its high computational demand, which was incompatible with our hardware resources. The results from the paper were used instead.
- Modifications were made to ensure the algorithm runs efficiently on our system while maintaining the core integrity of the ensemble learning framework.
- Changes were implemented in the plotting functions to replace numerical labels with correct element names under the respective features, improving the clarity and interpretability of the visualizations.

With these modifications, we maintained the model's predictive robustness.

6 Conclusions

6.1 Article Conclusions

The article concludes that the proposed ensemble learning model, optimized with GWO, effectively predicts COVID-19 mortality risk using a minimal feature set. This approach balances accuracy and efficiency, offering a practical solution for clinical applications.

6.2 Our Conclusions

Our findings confirm the article's conclusions that the proposed methodology is robust, efficient, and adaptable to similar predictive challenges. The ensemble learning approach demonstrates high performance with a reduced feature set.

A key observation in our study was the swapped importance of the features Age.1 and OsSats in the ERT model for both the 20- and 10-feature ensemble models. This indicates that the importance of these features is very similar, and under different conditions, the model prioritizes one over the other. This finding underscores the need to consider the context and conditions when interpreting feature importance in predictive models.

Future work could address the limitations of Grey Wolf Optimization (GWO), particularly its computational demand, by exploring hybrid optimization strategies that balance efficiency and performance.