Stage 2

Machine Learning Course

Major: Cloud Computing and Cybersecurity (Group CCC1)

Supervised By:

Nédra MELOULLI
Christophe RODRIGUES

Written By:

Alex QUILIS VILA
Carolina ROMERO DELGADO
Livia GAMERO HAMMERER

Date Last Edited: November 21, 2024

Contents

Chapter 1

Previous methods and limitations

1.1 Previous Methods

- Logistic Regression, Decision Tree, and Random Forest were implemented to predict heart disease.
- The Decision Tree was identified as the best-performing model with satisfactory metrics.

1.2 Limitations

However, many of these models struggle with class imbalance, leading to poor predictive performance for minority classes.

- Decision Tree risked overfitting due to its simplicity, even with limited depth.
- Random Forest provided competitive performance but required higher computational resources.
- Logistic Regression was insufficient for capturing non-linear relationships.

Chapter 2

Improvement Assumptions

To address these limitations, more advanced algorithms were considered, including K-Nearest Neighbors (KNN), LightGBM, CatBoost and XGBoost with the following assumptions:

- Advanced models can better handle complex relationships in the dataset.
- Ensemble methods can combine the strengths of multiple models to improve overall performance.

Chapter 3

Problem formalisation and methods

3.1 New Algorithm description

- K-Nearest Neighbors (KNN) A non-parametric algorithm that classifies a data point based on the majority class of its nearest neighbors.
- LightGBM A gradient-boosting framework optimized for speed and performance with categorical data support.
- CatBoost A gradient-boosting algorithm designed specifically to handle categorical features efficiently.
- XGBoost An optimized gradient-boosting framework that uses parallel processing, tree pruning, and regularization techniques to enhance performance.
- Ensemble Model: Techniques that combine multiple base models to improve predictive performance compared to using a single model. This approach leverages the strengths of each individual model while mitigating their weaknesses, resulting in a more robust performance. Two common ensemble methods are the Voting Classifier and the Stacking Classifier, both of which are implemented in the code for performance comparison.
- Voting Classifier: Ensemble method that combines multiple base models and makes a final prediction based on their individual predictions. There are two types of voting:
  - Hard Voting: The class that gets the most votes (i.e., the majority prediction) is chosen as the final output.
  - Soft Voting: This method uses the predicted probabilities from each base model and averages them to calculate the final class. Soft voting

typically provides better results as it takes into account the uncertainty in the predictions.

The Voting Classifier is implemented with four base models: Random Forest, K-Nearest Neighbors (KNN), LightGBM, and XGBoost. Soft voting is used, meaning the model combines the probability predictions from each base model to make the final decision. The model is then trained on the training data and evaluated using accuracy and ROC AUC scores.

- Stacking Classifier: Ensemble technique where multiple base models are trained and their predictions are used as input to a final model (called the meta-model). The meta-model learns how to optimally combine the predictions of the base models. In this case, the base models are Random Forest, K-Nearest Neighbors (KNN), and LightGBM, and the meta-model is XGBoost. The meta-model is trained to learn the best way to combine the predictions from the base models to improve the final prediction. The model is also trained on the training data and evaluated using accuracy and ROC AUC scores.

3.2 Limitations

- KNN: High computational cost for large datasets.
- LightGBM, CatBoost and XGBoost: Require expertise in hyperparameter tuning for optimal performance.
- Ensemble approaches can increase complexity and runtime.

Chapter 4

Methodology

4.1 Data preprocessing (as in the previous stage)

- Addressed class imbalance with SMOTE.
- Scaled numerical features for algorithms like KNN.
- Encoded categorical variables for gradient-boosting methods.

4.2 Training and testing split

- 80% training, 20% testing.
- Scaled numerical features for algorithms like KNN.
- Applied cross-validation for robust evaluation.

4.3 Model implementation and comparison

- Evaluated each advanced model individually.
- Built an ensemble model combining the top-performing models.

4.4 New Algorithm Implementation and Hyperparameters

- **KNN**:
  - **Key Hyperparameter**: `n_neighbors` set to 5, representing the number of neighbors considered.

  - **Preprocessing**: Features normalized using `StandardScaler` to ensure consistent distance calculations.

- **LightGBM**:
  - **Key Hyperparameter**: `num_leaves` set to 31 (for optimal trade-off between complexity and speed).
  - **Learning Rate**: Set to 0.05 for better performance with fewer iterations.
  - **Preprocessing**: Categorical features were handled automatically by LightGBM, no manual encoding required.

- **CatBoost**:
  - **Key Hyperparameter**: `iterations` set to 1000, `learning_rate` set to 0.1 for optimal trade-off between speed and performance.
  - **Preprocessing**: Categorical features automatically detected and processed without manual encoding.

- **XGBoost**:
  - **Key Hyperparameter**: `max_depth` set to 6, `learning_rate` set to 0.1 for fast convergence.
  - **Regularization Parameters**: `alpha` set to 0.5 to prevent overfitting, `lambda` set to 0.5 to control model complexity.

Chapter 5

Results

Models	Accuracy	Precision	Recall	F1-Score	ROC AUC
KNN	0.7407	0.1778	0.5160	0.2644	0.6395
LightGBM	0.7230	0.1984	0.6796	0.3071	0.7035
CatBoost	0.7443	0.2025	0.6231	0.3056	0.6897
XGBoost	0.7497	0.2025	0.6031	0.3032	0.6837
Voting classifier	0.7454	0.2028	0.6204	0.3057	0.7729
Stacking classifier	0.8178	0.1984	0.3345	0.2491	0.7518

Table 5.1: Model Results

5.1 Metrics

- Accuracy Provided the overall correctness of predictions.
- Precision Measured the relevance of positive predictions.
- Recall Evaluated the model's ability to capture true positives.
- F1-Score Balanced metric for precision and recall.
- ROC-AUC Assessed the performance of models by comparing true positive rates against false positive rates.

5.2 Overfitting

The models demonstrated minimal signs of overfitting, with the ensemble models showing more balanced performance across the evaluation metrics compared to the individual models. While the Stacking classifier performed the best overall in terms of accuracy, the Voting classifier achieved the highest ROC AUC score, indicating strong generalization across various metrics.

5.3 Evaluation Comparison with Previous Solutions

The new models, particularly the Voting and Stacking classifiers, showed significant improvements over the previous basic models. For instance, the Voting Classifier improved accuracy by approximately 1% and ROC AUC by over 13% compared to XGBoost, while the Stacking Classifier achieved even better results with an accuracy of 0.818 and ROC AUC of 0.752, which are noticeably higher than the individual models. Although the Stacking Classifier achieved lower precision, its overall performance in terms of accuracy and ROC AUC still made it the most effective model.

Chapter 6

Discussion and Conclusion

6.1 Discussion

- The advanced models, especially XGBoost and LightGBM, demonstrated the ability to handle complex data patterns, but the overall best performance came from combining models into an ensemble approach.
- The Voting classifier, while not achieving the highest accuracy, outperformed other models in terms of ROC AUC, showing that combining classifiers can significantly boost performance on certain metrics.
- The Stacking Classifier, despite a lower F1-score, delivered the best balance of accuracy and ROC AUC, making it the top model for this task.

6.2 Conclusion

From the results, we observe that the ensemble models (Voting and Stacking Classifiers) outperformed the individual models, particularly in terms of ROC AUC and accuracy. The Voting Classifier showed the highest ROC AUC score, while the Stacking Classifier led in accuracy. While the XGBoost and LightGBM models were strong, the ensemble approach provided a better overall result, highlighting the effectiveness of combining multiple classifiers for this prediction task.