Jingyu Wang

04/15/2023

Biostat 626 Midterm 1: Problem Set

1. Complete the Task 1 and 2 and submit your classification results via Canvas:

Here are screenshots of I submitted my classification results for task 1 and task 2 on canvas:

图形用户界面, 文本, 应用程序

描述已自动生成 图形用户界面, 文本, 应用程序

描述已自动生成

1. Set up a Github repository and upload all your code used for training, evaluation, and generating results of test data. Provide the url to your Github repository as the answer to this question.

My Github url: <https://github.com/wjingyu06/jingyu-Biostat-626-midterm1.git>

1. Write a text file (name the file "README.md") to provide necessary instructions, so that other people can reproduce all your results.

url for README.md: <https://github.com/wjingyu06/jingyu-Biostat-626-midterm1/blob/main/README.md>

1. Describe your baseline algorithm and provide necessary tables and/or figures to summarize its performance based on the training data.

Task 1 baseline algorithm performances based on the training data:

For task 1's binary classification performance based on the training data, I employed four baseline algorithms: Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Decision Tree (DT), and Linear Discriminant Analysis (LDA). The following Table 1 displays the performance of each model using the training data:

|  |  |
| --- | --- |
| **Models** | **Accuracy** |
| SVM | 0.998713 |
| KNN | 0.999571 |
| LDA | 0.999571 |
| DT | 0.990991 |

**Table1: Binary Classification Baseline Algorithms ‘Performances on Training Data**

From Table 1, we can observe that KNN and LDA have the best performance in classifying time window activities into static and dynamic categories, while the Decision Tree has the lowest accuracy, albeit still achieving 0.990991. KNN's non-parametric properties and its ability to handle the classification led to high performance in task 1. Additionally, LDA's compatibility with the data's distribution and the algorithm's assumptions contribute to its outstanding performance.

Task 2 baseline algorithm performances based on the training data:

For task 2’s multiclass classification performance based on the training data, I employed five baseline algorithms: Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Decision Tree (DT), and Random Forest (RF). The following Table 2 shows the performance of each model using the training data:

|  |  |
| --- | --- |
| **Method** | **Accuracy** |
| SVM | 0.9734020 |
| KNN | 0.9884170 |
| LDA | 0.9785500 |
| DT | 0.8687259 |
| RF | 0.9794080 |

**Table2: Multi-class Classification Baseline Algorithms ‘Performances on Training Data**

From Table 2, we can observe that KNN has the best performance, followed by Random Forest, while Decision Tree has the lowest accuracy among all five models. KNN's superior performance can be attributed to its non-parametric properties and adaptability to complex data relationships, which are particularly beneficial for multi-class classification. Additionally, Random Forest, as an ensemble method, reduces overfitting by leveraging multiple Decision Trees, consequently resulting in high performance. Finally, the Decision Tree's lowest accuracy of 0.8687259 is due to its tendency to overfit the training data.

1. Describe your final algorithm and provide necessary tables and/or figures to summarize its performance based on the training data.

Task 1 final algorithm performances based on the training data:

Based on the performance of baseline algorithms in Table 1, I decided to use KNN and LDA to perform final prediction based on the whole training dataset, the following Table 3 shows the performance of final model using the whole training data:

|  |  |
| --- | --- |
| **Method** | **Accuracy** |
| KNN | 0.999571 |
| LDA | 0.999571 |

**Table 3: Binary Classification Final Algorithms ‘Performances on Training Data**

Since KNN has better performance than LDA, therefore I decided to use KNN as my final model, also on the leaderboard KNN’s accuracy achieved at 0.997, further support its advantages. KNN's remarkable performance in binary classification can be attributed to several reasons. For instance, the KNN model can effectively capture small variations in the dataset better than other models, and choosing an appropriate value for K can help improve the model's performance.

Task 2 final algorithm performances based on the training data:

Based on the performance of baseline algorithms in Table 2, I decided to use KNN and LDA to perform final prediction based on the whole training dataset, the following Table 4 shows the performance of final model using the whole training data:

|  |  |
| --- | --- |
| **Method** | **Accuracy** |
| KNN | 0.9884170 |
| RF | 0.9794080 |

**Table 4: Muti-class Classification Final Algorithms ‘Performances on Training Data**

Although KNN has a higher accuracy (0.9884170) than RF (0.9794080), I decided to use RF as my final model for multi-class classification due to its strengths. Also RF has advantages which leverages multiple decision trees to reduce overfitting and provide a more robust classification. By aggregating predictions from multiple trees, RF captures complex decision boundaries and improves generalization. Both KNN and RF excel in multi-class classification due to their ability to adapt to data structures and handle complex data relationships.

1. Use a figure or a table to show your leaderboard performance. Describe your efforts to improve the performance.

Task 1 Binary Classification Leaderboard Performance:

|  |  |
| --- | --- |
| **Submission** | **Accuracy on Leaderboard** |
| 1st Submission (KNN 7467) | 0.997 |
| 2nd Submission (LDA 746766) | 0.997 |

**Table 5: Binary Classification Leaderboard Performance**

I submitted the task 1 prediction results to the leaderboard twice. The first submission, with ID 7467 (Appendix 1), used the KNN model, which achieved an accuracy of 0.997 on the leaderboard. As LDA had the same accuracy as KNN on the training data, I fine-tuned the LDA model and submitted its prediction results for the second time. The ID of the second submission was 746766 (Appendix 2); however, the accuracy remained at 0.997. Consequently, I decided to use KNN as my final model and result.

Task 2 Multi-class Classification Leaderboard Performance:

|  |  |
| --- | --- |
| **Submission** | **Accuracy on Leaderboard** |
| 1st Submission (KNN 7467) | 0.877 |
| 2nd Submission (RF 746766) | 0.929 |

**Table 5: Multi-Class Classification Leaderboard Performance**

I submitted the task 2 prediction results to the leaderboard twice as well. The first submission, with ID 7467 (Appendix 3), used the KNN model and achieved an accuracy of 0.877 on the leaderboard. However, this performance was not as good as that observed in the training dataset. Consequently, I decided to use the Random Forest model for the second submission. After tuning the parameters in Random Forest, I submitted the final prediction with ID 746744 (Appendix 4), resulting in an improved leaderboard accuracy of 0.929. Therefore, in multi-class classification, Random Forest algorithm has better performance than the KNN algorithm.

1. Comment on your final results and potential ways to further improve the classification accuracy.

The final results I obtained for Task 1 used the KNN model with an accuracy of 0.997, and for Task 2, the Random Forest model with an accuracy of 0.929. In the future, I believe I should conduct more data inspection and implement feature engineering to improve classification accuracy. For instance, I could apply PCA to reduce the number of features. Additionally, I could experiment with the dataset using a wider range of machine learning models and perform more effective hyperparameter optimization, such as Grid Search. Furthermore, I should allocate more time for myself to complete the task and strive to enhance the classification accuracy.

Appendix

图片包含 表格

描述已自动生成

**Appendix 1:** Screenshot of 1st submission on binary classification

表格

描述已自动生成

**Appendix 2:** Screenshot of 2nd submission on binary classification

图片包含 表格

描述已自动生成

**Appendix 3:** Screenshot of 1st submission on multi-class classification

表格

描述已自动生成

**Appendix 4:** Screenshot of 2nd submission on multi-class classification