626 Midterm1 Problem Set

Problem 1:

Results have been submitted through canvas and grade has been received through leaderboard. The file names are: binary\_1957.txt and multiclass\_957777.txt.

Problem 2: <https://github.com/scarletthesijia/Biostat626_midterm1>

Problem 3:

The README file is uploaded on Github. It can be found from the link above.

Problem 4 – Problem 7 is written separately for Binary and Multiclass classifiers.

**Binary:**

Problem 4:

In the provided dataset, there are 12 activity codes which were classified as static or dynamic. In order to classify these activity codes into static and dynamic, activities with code 4,5,6 were classified as static(0) and 1, 2, 3, 7, 8, 9, 10, 11, 12 were classified as dynamic(1). The classified training data was used for model fitting.

An initial model was constructed using a generalized linear model (GLM) with a binomial family, which is a binary logistic regression model commonly used to model the probability of the binary response variable given a set of predictor variables. The initial model included all variables.

To evaluate the initial model's performance on the training data, the data was split into a training dataset with 70% of the original training data and a testing dataset with 30% of the original training data. The true activity codes in the testing dataset were removed and saved for future accuracy performance calculation. The initial predicted activity codes were predicted using the initial model on the testing dataset and compared with the true activity coding. The accuracy performance of the initial model was found to be 0.997 based on the training data and 0.991 on the leaderboard.

|  |  |  |
| --- | --- | --- |
| Model | Accuracy on Training Data | Accuracy on Leaderboard |
| Initial model | 0.997 | 0.991 |

Table1: Initial model prediction accuracy for binary classifier.

Problem 5:

The final algorithm used for this study remains a generalized linear model (GLM) with a binomial family. All variables were included. The model achieved a perfect accuracy score of 1.000 on both the training data and leaderboard. The methods utilized to improve the initial model's performance will be explained in problem 6.

|  |  |  |
| --- | --- | --- |
| Model | Accuracy on Training Data | Accuracy on Leaderboard |
| Final model | 1.000 | 1.000 |

Table2: Final model prediction accuracy for binary classifier.

Problem 6:

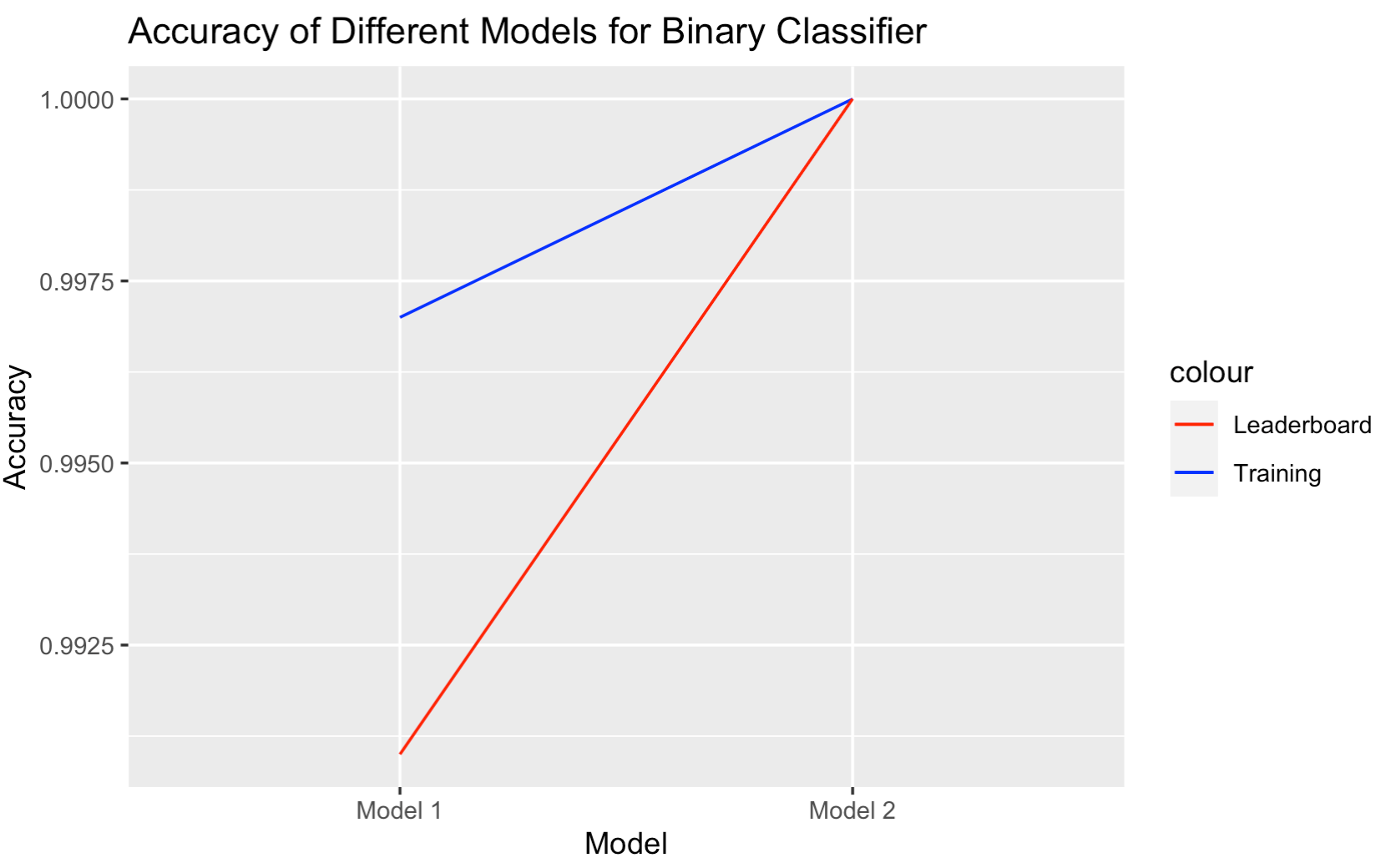
Efforts to improve the performance:

After reading the data dictionary and the README file, I realized that my initial binary category was not correct. Only activities 1, 2, and 3 should be considered as dynamic, while other activities are all static. So, I retrained the model using the corrected categories and built a new binary classification model.

The prediction accuracy for all models used to build binary classifier are shown in table 3 and visualized on plot 1.

|  |  |  |
| --- | --- | --- |
| Model | Accuracy on Training Data | Accuracy on Leaderboard |
| Initial model | 0.997 | 0.991 |
| Final model | 1.000 | 1.000 |

Table3: Prediction accuracy for all models for binary classifier.



Plot1: Prediction accuracy for all models for binary classifier

Problem 7:

The final model had an accuracy of 1.000 on the training data and the leaderboard, which is a good result. This indicates that the model is performing well in predicting the activity category. I need to be more careful on reading data dictionary and learning tasks when building the model. In order to further improve the classification accuracy, I could try using more advanced machine learning algorithms or techniques, such as neural networks or ensemble methods.

**Multiclass:**

Problem 4:

In order to fit the multiclass classifier, activity code 7,8,9,10,11,12 are classified as static postural transition (7). The classified training data was used for model fitting.

For this initial model, a support vector machine (SVM) with a radial kernel was used for multiclass prediction. All variables were included in the model fitting process. The preparation of the testing dataset from the original training dataset followed the same procedure as described for the binary classifier. The accuracy of this model was 0.915 on training data and 0.950 on the leaderboard. These accuracy are shown on the table4.

|  |  |  |
| --- | --- | --- |
| Model | Accuracy on Training Data | Accuracy on Leaderboard |
| Initial model | 0.915 | 0.950 |

Table4: Initial model prediction accuracy for multiclass classifier

Problem 5:

To enhance the performance of the model, control was added through cross-validation, and the model was tuned by finding the optimal cost and kernel. The best-performing model was obtained when the cost was set to 0.1 and the kernel was linear, with a cross-validation value of 10. To further improve the model, a random forest was added to select important variables and these selected variables were used in the SVM model. The details of this improvement process will be discussed in problem 6.

Finally, based on the selected variables from the random forest, the final algorithm was built using the SVM model with the control set to cv=10 and cost=0.1. The prediction accuracy is also shown on Table5.

|  |  |  |
| --- | --- | --- |
| Model | Accuracy on Training Data | Accuracy on Leaderboard |
| Final model | 0.984 | 0.958 |

Table5: Final model prediction accuracy for multiclass classifier

Problem 6:

Efforts to improve the performance:

The initial SVM model was improved by incorporating control measures to enhance its predictive performance. A tuning process was performed to identify the optimal combination of hyperparameters for the SVM model. The cost hyperparameter was selected from a range of values, including 0.1, 1, and 10, while the kernel was chosen between linear and radial. The train control parameter was also tuned, with the method selected from among cv, boot, and repeatedcv. Number was chosen from 5, 10, and 20. The best combination identified during the tuning process was an SVM model with a linear kernel, using cv as the train control method with 10 as the number parameter. The accuracy of this tuned SVM model was 0.987 on the training data and 0.956 on the leaderboard.

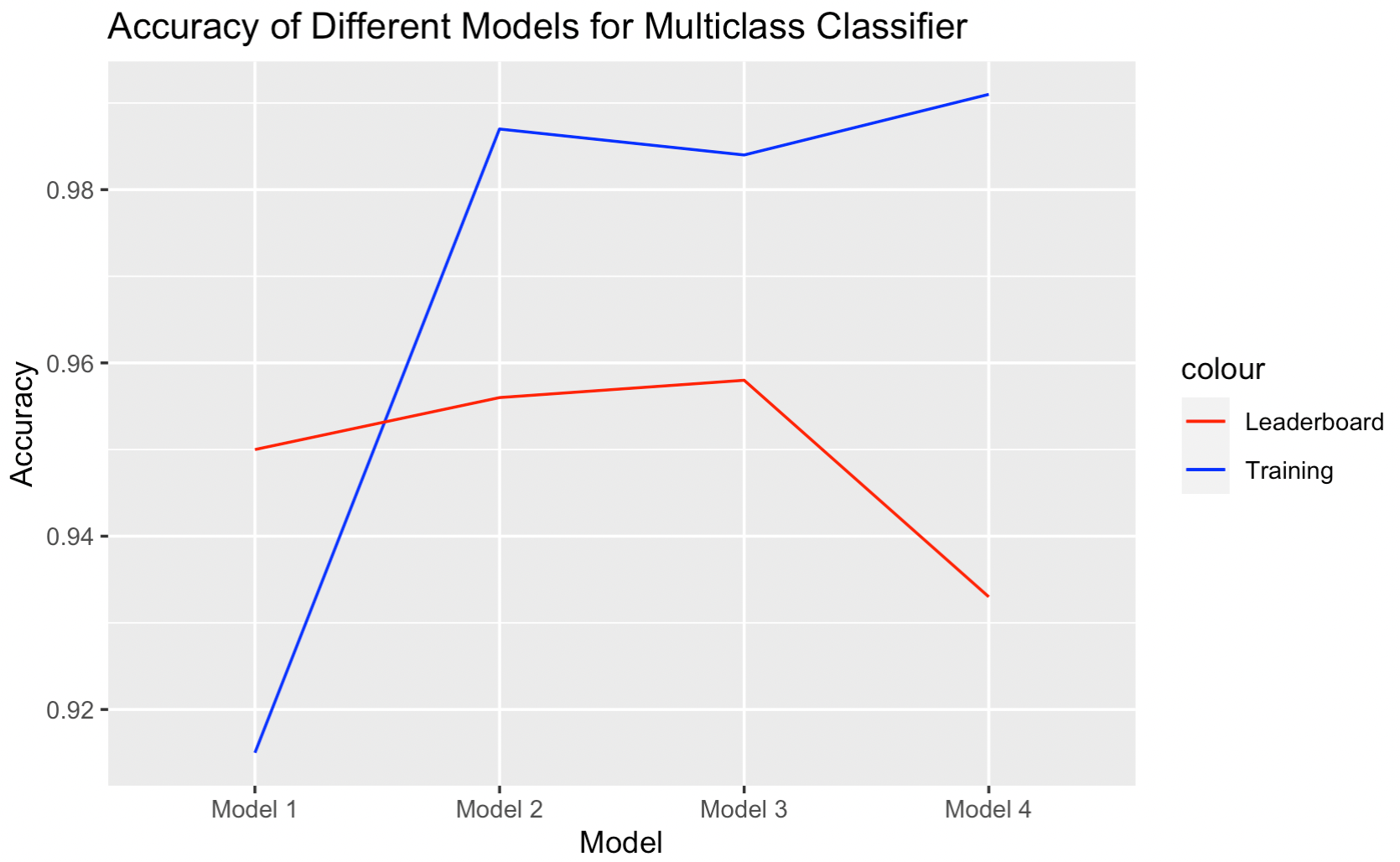
To further improve the model, a random forest algorithm was used to select features from all variables. The number of trees used is 500. And the top 500 variables were selected based on the MeanDecreaseAccuracy value. The tuned SVM model was then fitted again using the selected variables. The resulting accuracy was 0.984 on the training data and 0.958 on the leaderboard.

Additionally, a neural network was used to build a classifier, with the model being tuned to find the best inputs. Different combinations of the learning rate, batch size, activation, and dropout rate were tested to identify the optimal combination for prediction. The accuracy of this neural network model was 0.991 on the training data and 0.933 on the leaderboard. A plot is also included here for visualization of the accuracy change of different models I used. The model 1 is the Initial SVM model. Model 2 is the SVM with control. Model 3 is the SVM with control and random forest. This model has the best performance on leaderboard and relatively good performance on training data. So, this is the final model. Model 4 is the model built based on Neural Network, which does not have good performance on leaderboard so this model is not considered for final model.

The prediction accuracy for all models used to build multiclass classifier are shown in table 6 and visualized on plot 2.

|  |  |  |
| --- | --- | --- |
| Model | Accuracy on Training Data | Accuracy on Leaderboard |
| Initial SVM model | 0.951 | 0.950 |
| SVM with control | 0.987 | 0.956 |
| SVM with control and random forest | 0.984 | 0.958 |
| Neural Network | 0.991 | 0.933 |

Table6: Prediction accuracy for all models for multiclass classifier



Plot2: Prediction accuracy for all models for multiclass classifier

Problem 7:

There is a noticeable improvement from the initial model to the final model. The overall prediction accuracy is 0.958 on the leaderboard, which is a relatively high performance.

There are some potential improvements that can be applied to the model to increase accuracy. The performance of SVM models can be sensitive to the scaling of the input data. Consider standardizing the data before training the model to ensure that each feature contributes equally to the classification decision. Even though I have already tried to tune the model, I can still try to use a more extensive grid search or try different combinations of hyperparameters.

The R code for the final models can be found in the github R section.

The related r code for model improvement can also be found in github in rmd files named as Binary\_Model\_Development.Rmd and Multiclass\_Model\_Development .Rmd.