SVM for Classification : Ionosphere

Krishna Kalyan

Introduction

The aim of this assignment is to to perform SVM classification on some data set of our choice and compare standard multivariate classification methods like `lda`, `glm` to `svm`. Finally we need to draw some conclusions based on results. For this assignment we have chosen Ionosphere database donated by Vince Sigillito.

This radar data was collected by a system in Goose Bay, Labrador. This system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kilowatts. See the paper for more details.The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those that do not; their signals pass through the ionosphere.

This is a binary classification task.

Approach

Our apparoach will broadly be based on the following sequence below:

- Summary Statistics and Pre-Processing
- Model Fitting
- Cross Validation
- Evaluation
- Conclusion

Pre-Processing

Our data contains `351` observations and `35` attributes. The second attribute `V2` was removed as it only had zeros. All `34` attaributes are continuous and the thirty fifth `V35` attribute is either "good" or "bad" according to the definition summarized above.

Model Fitting

Initially, We will train our data based on standard evaluation methods like Naive Bayes classifier, Recursive partitioning for classification, Random Forest, Neural Networks (With single hidden layer), Linear Discriminant Analysis, Generalized Boosted Regression Models. All models mentioned above were trained using the default parameters.

We will also compare standard models with `SVM` from the `kernelab` package. Hopefully `SVM` with one of the kernels will give us a high accuracy and high F1 score. Once we have the `SVM` with best kernel, we will tune its hyperparameters using grid search. This will enable us to obtain a very good model and its tuned hyper parameters.

Cross Validation

For this assignment we have decided to use 5 fold cross validation. Usually kfold crossvalidation is used to prevent overfitting and gives an insight on how the model will generalize to an independent dataset.

Before implementing the kfold cross validation stratergy we make sure that we shuffle input data before splitting them into `training` and `test` set. The split ratio for `training` to `test` set is `80-20` percent. This `test` set contains labels.

This `training` set is again split for `5-fold` cross validation into traning and validation set.

Evaluation

Since its a binary classification task we will be considering two evaluation approaches below:

- F1 Score
- Accuracy

Runtime to train each model in my computer will also be considered as an important criteria. `Accuracy` and `F1 Score` have similar plots as observed below.

F1 Score

In statistical analysis of binary classification, the F1 score is a measure of a test's accuracy. It considers both the precision `p` and the recall `r` of the test to compute the score.

```
F1 = p * r / (p + r)
```

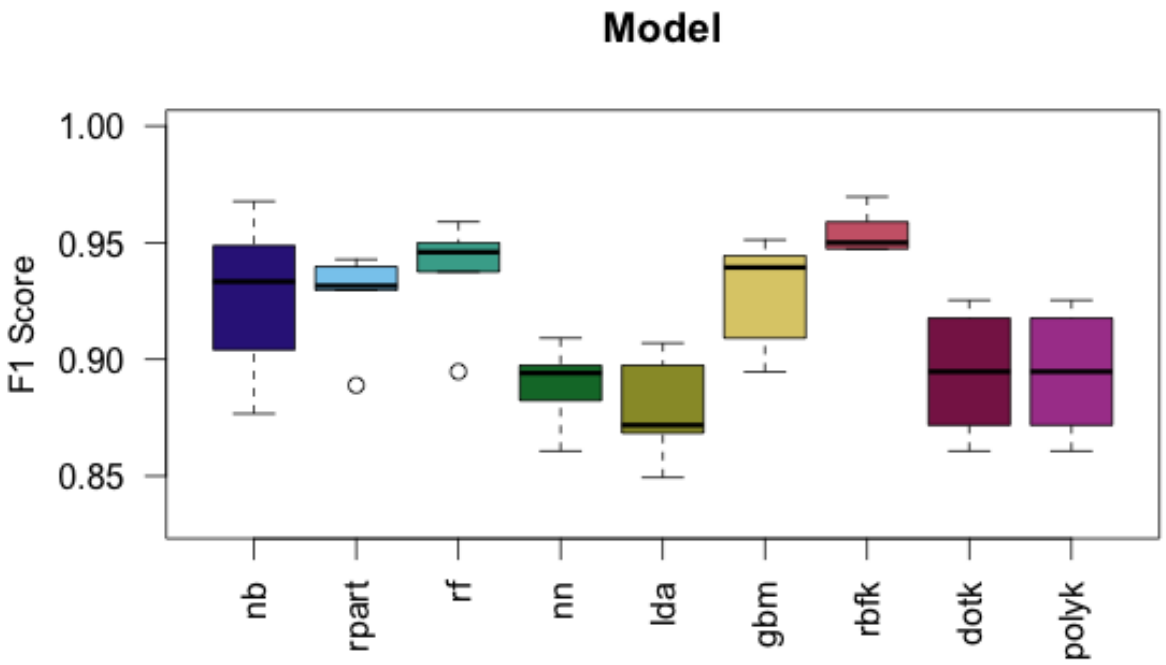

Figure 1:

In this case `SVM` with `RBF Kernel` had the best performance with average `F1 Score` over `5` folds being `0.954` and the worst being `Linear Discriminant Analysis` with average its being `0.878`.

Accuracy

To compute accuracy the set of labels predicted for a sample must exactly match the corresponding set of labels.

In this case `SVM` with `RBF Kernel` had the best performance with average `Accuracy` over `5` folds being `0.939` and the worst being `Linear Discriminant Analysis` with average accuracy being `0.832`.

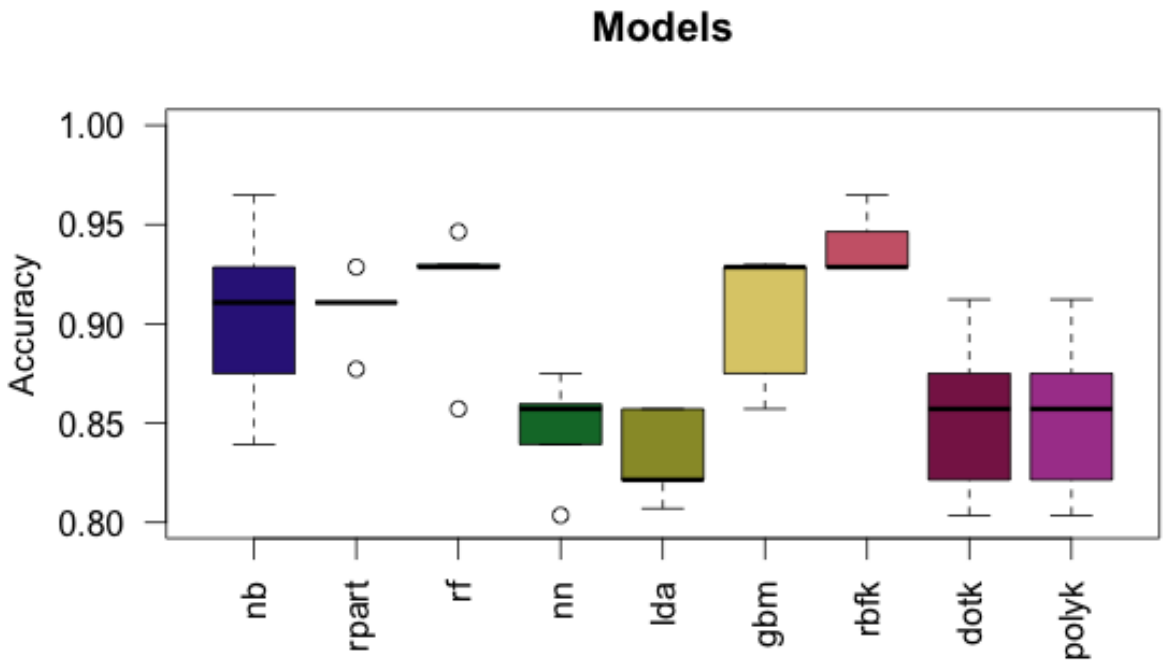
Models
Accuracy
1.00
0.95
0.90
0.85
0.80
nb
rpart
rf
nn
lda
gbm
rbfk
dotk
polyk

Figure 2:

Computational Time

We can observe below that `Random Forest` takes more time compared to other models. They also seem to have more variation in the training time. Both the models were trained using default parameters.

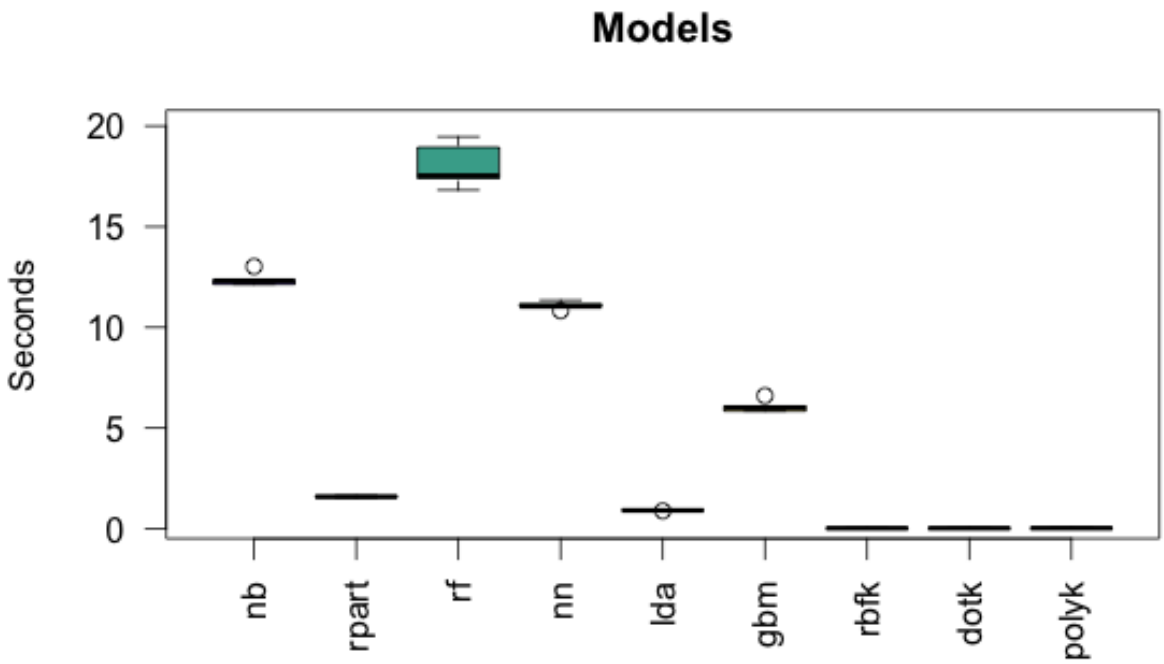

Figure 3:

In this case `SVM` with `RBF Kernel` had the best performance with average time taken to train over `5` folds being `0.0246` seconds with worst being `Random Forest` with average time over `5` folds being `18.03` seconds.

Conclusion

Based on the experiments we can clearly see that `SVM` with `RBF` kernel has the best performance with repect to `F1 Score`, `Accuracy` and `Time Taken`. Due to time constarints I did a very small grid search for a best `sigma` value keeping `C` constant and found that the best sigma is `0.1`.

Using the best hyper parametes obtained with original `training` and `test` data we observe an accuracy of `0.97`, `F1 Score` of `0.97` with total time taken to train being `0.647`.

We know that `SVM` does not scale well and is difficult to parallize. However in this experiment we can certainly conclude that `SVM` with `RBF` kernel outperforms most of the models.

Future Work

I would like to explore the state of art in `SVM Kernels`. Also check out onging research on `SVM` parallelization. I know my code is a bit verbose. I would perfer keeping it simple, readable and easier to debug than build functions which would make my code complex.