Machine Learning- Project 2019

Hai Long, Le (18200524)

Schools of Mathematics and Statistics- UCD

STAT30270

Statistical Machine Learning

Abstract

Machine Learning is not just a simple step of applying the well-known algorithms to the Dataset to get the prediction accuracy. More than that, in order to get the best out of, Machine Learning requires multiple other steps such as: Feature Selection, Pre-Processing, Algorithm Evaluation, and Parameters Tuning. In my project, I will apply these steps to Backpain Dataset to get the best performance and make the Machine Learning even much more powerful. The main objective of this project is to build the Predictive Model for Binary Classification of Backpain Dataset to optimize the clinical outcomes. Using different of methods to train, evaluate, and discuss to find out the most effective method. Building the powerful Machine Learning Classifier is important but it is not the only factor of this project. I will also apply the knowledges I have learnt throughout this module to get other interesting and dynamic features to understand more about how the algorithms work such as the Variable Importance, Resampling techniques, Bias-Variance Trade-Off, Model Comparison and Selection, etc. I will focus on 3 Classification methods we spent quite time in this module to train my model which are: Ensemble, Bagging, and Kernel. Moreover, I will perform the Tuning Parameters to improve the performance baseline models in order to have the best prediction on the unseen data.

I. Introduction.

The Dataset "Backpain" we use for this project have 380 observations, 31 Predictors with Target is in binary class (Nociceptive/ Neuropathic). This Dataset is the collection of both categorical and numerical predictors. We are required to build the Binary Classifier to optimize the clinical outcomes by predicting either that person has Nociceptive or Neuropathic.
The main objective of this project is to apply the knowledges I have learnt in this Module to build the Machine Learning and get the most out of it, and evaluate using multiple performance metrics. First of all, I will split my data into 3 parts: Train Set, Validation Set, and Test Set to allow how much of the data my Machine Learning model are allowed to see. After that, I will break my projects into 6 small tasks:

- ***Task 1*: Exploratory-Data-Analysis**. This task will have me understand more about my data using Descriptive analysis.
- ***Task 2*: Transforming Data**. At this step I will transform my data to more standardized structure to help the machine learning model easily uncover my data
- ***Task 3*: Apply Algorithm to Baseline Models and Evaluate**. At this step, I will apply 3 different methods(Ensemble, Bagging, Kernel) to train my baseline model and then evaluate them using Validation set.
- ***Task 4*: Improve Accuracy**: I will apply some Feature Selection methods and Pre-Processing to the Baseline models and evaluate Validation set to check for improvements.
- ***Task 5*: Tuning Parameters:** I will select 2 best model from Baseline models to tune the parameters to improve the results.
- ***Task 6:* Finalize Model:** I will select the "best" model to predict for unseen data in Test set and present the classification results.

Along with the small tasks above, I will answers some interesting questions relate to my machine learning model for this dataset. I would like to know both Bias and Variance of each Baseline models on the Train Set. Also, I believe that I would be helpful if I can know the prediction of my Baseline models on the Validation Set (underfitting, well-fitting, or overfitting). As I will apply the Pre-Processing to my models, I am interested in how beneficial the Pre-Processing steps will contribute to the overall accuracy of Validation Test. In this step, I will make the decision based on the Variable Importance given by Random Forest, and try to standardize and reduce data dimensions the data by Principle Components. Furthermore, I will try to figure out the best parameters of top 1 baseline models to improve the accuracy for the Test Set. The dataset is balanced data so I will access the performance of my models using Accuracy, Specificity, Sensitivity, and ROC curve.

II. Methods.

1. *Data Splitting and Resampling Techniques.*

In this project, I divided my dataset into 3 different sets which are Train Set, Validation Set, and Test Set. Train Set is the set of data to train/ fit the models. The models can see the whole data to learn from it. Validation Set will be used to evaluate the model fit based on the Train Set. Validation Set will provide unbiased evaluation on the models. Using Validation Set, I will access the ability of model fit whether overfitting or underfitting. The Test Set is to evaluate the final model fit on the unseen data to confirm the prediction power. I randomly selected 10% of my data as Test Set, 20% of remaining data as Validation Set, and the rest is Train Set.

Similar to Bootstrap we have learnt in this Module, "Repeated Cross-Validation" is a powerful resampling technique to estimate the accuracy of the model. In my project, with 10-fold and each fold will be repeated the estimation for 3 times. In other words, it will divide my data into 10 folds, 9 folds will be use to train the model and then test the error rate with 1 fold left, and it will be repeated for 3 times. The error rate of estimations will be averaged out to have the expected error of the whole model. Using this Repeated CV not only gives the better estimation of model performance but also reduce the Bias to improve the Prediction Accuracy.

2. *Pre-Processing (Standardize and Principle Component Analysis (PCA)).*

I would not apply any Pre-Processing steps to my Baseline models at first and then apply Pre-Processing to my models to see if It can improve the performance of my models. In my project, I will use 2 Pre-Processing techniques which are: Standardize Data and PCA.

Standardize Data will make the attributes have Mean of 0, Variance of 1. In other words, Standardized data will treat all the variables in our data equally. It also supports PCA.

PCA is a multivariate technique to deal with highly correlated variable, and to perform the dimensionality reduction with minimal loss of the information. PCA will extract the important information of the whole dataset and will express with fewer new variables. After performing the EDA of the dataset using Pearson Correlation for Continuous Variables, I have figured out that there are some highly correlated variables so It may useful to apply PCA to data.

3. Variable Selection using Variable Importance.

Variable Selection is an important step of model building. The Variable Selection is expected to improve the prediction accuracy by eliminating the unnecessary variables to remove the noise and improve computation efficiency. Building the "quick random forest" with 500 trees and accessing the Variable Importance, I will remove few least contributing Variables based on Mean Decrease Gini. The Predictors with low Mean Decrease Gini will not appear in many splits in tree-based models.

4. Machine Learning Algorithms.

Random Forest: is a powerful classification method using tree-based model with bootstrapping to make it more stable and reliable. Random Forest will take the bootstrap of the data, build the classification tree at each split using only the random subset of variables and then test with Out-Of-Bag samples.

Bagging: is also known as Bootstrap Aggregation. Bagging works quite similar to Random Forest. The main difference between these 2 methods is Bagging does not create the subset of variable at random. Theoretically, Random Forest is expected to be more powerful than Bagging.

Boosting: is another ensemble technique to create a strong classifier from many weak learner. Weak learners are prepared on the training data using Weighting. At each iteration, the misclassified observations will be upweighted. After that, Boosting classifies the Testing data using the calculated weighting.

SVM: is the method classifies data using hyperplane. SVM is solved using the Constrained Optimization problem by finding the Maximum Margin Hyperplane, and applying Cost function. For Non-Linearity, It transform the data to high-dimension to separate the linear plane easier.

5. *Tuning Parameters.*

From the Baseline Models, we would choose the "best" model in term of Bias-Variance Tradeoff to tune the parameters. Tuning Parameters is to find the optimal parameters for algorithm to make the powerful machine learning perform even better. Each machine learning will have unique parameters to tune. The algorithms are parameterized are expected to perform better.

6. *Performance Measures.*

We have balanced class data, besides the Accuracy, I will access other performance metrics such as Balanced Accuracy, Sensitivity and Specificity. Because of Binary Classification, I would pay more attention on ROC Curve to have a deeper look into Sensitivity and Specificity.

III. Results.

1. *Exploratory Data Analysis.*

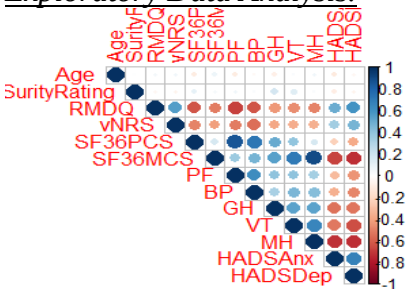

Figure 1.1 Correlation for Continuous

```
$ Gender          : Factor w/ 2 level
$ DurationCurrent: Factor w/ 6 level
$ PainLocation    : Factor w/ 6 level
$ Criterion2      : Factor w/ 2 level
$ Criterion4      : Factor w/ 2 level
$ Criterion6      : Factor w/ 2 level
$ Criterion7      : Factor w/ 2 level
$ Criterion8      : Factor w/ 2 level
$ Criterion9      : Factor w/ 2 level
$ Criterion10     : Factor w/ 2 level
$ Criterion13     : Factor w/ 2 level
$ Criterion19     : Factor w/ 2 level
```

Figure 1.2 Structure of Categorical Variable

2. *Baseline Models vs. Models after Pre-Processing and Variable Selection.*

As mentioned in previous sections, I would build Baseline Models using 4 different machine learning methods (Random Forest, Bagging, Boosting, SVM), and then apply Pre-Processing (Standardize, PCA), and VarImp of Random Forest with 500 trees to answer the question that if the Pre-Processing and Variable Selection would benefit our models. Moreover, I would train the models on Training Data and then test the Accuracy with Validation Data.

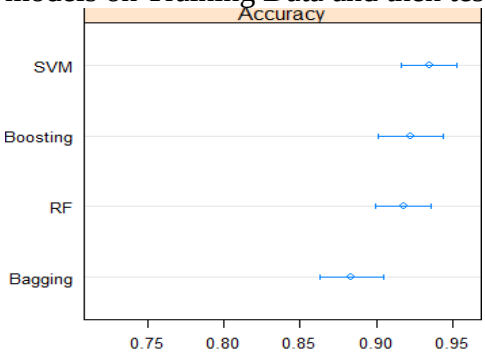

Figure 2.1 Accuracy of Baseline Models On Training Data.

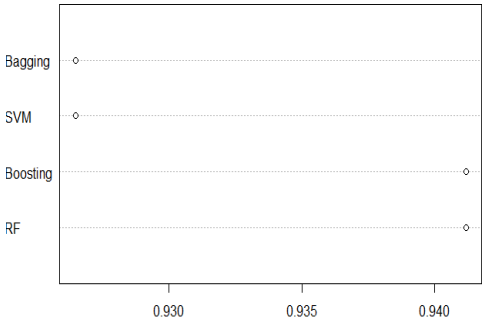

Figure 2.2 Accuracy of Baseline Models on Validation Data.

Figure 2.1 is the Accuracy (correctly classified) of Baseline Models on Training Data. Figure 2.2 is the Accuracy of Baseline Models on Validation Data. These two figures are to help us understand more about prediction ability of Baseline Models on seen data and unseen data.

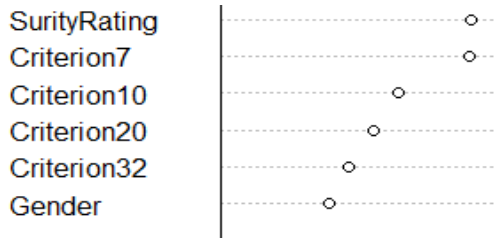

Figure 2.3 VarImp using Random Forest.

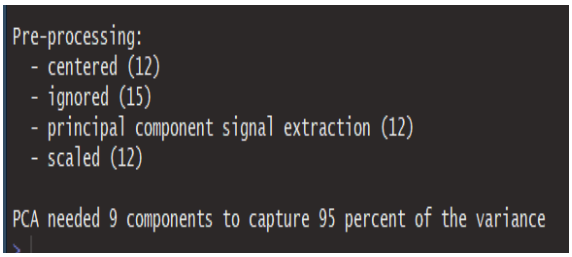

Figure 2.4 Pre-Processing Results

Figure 2.3 displays the Least important predictors given Variable Importance of Random Forest calculated based on the Mean Decrease Gini. Figure 2.4 displays the summary output of data after applying Standardize and PCA. These 2 figures provides us more information about how data is transformed after applying Variable Selection and Pre-Processing.

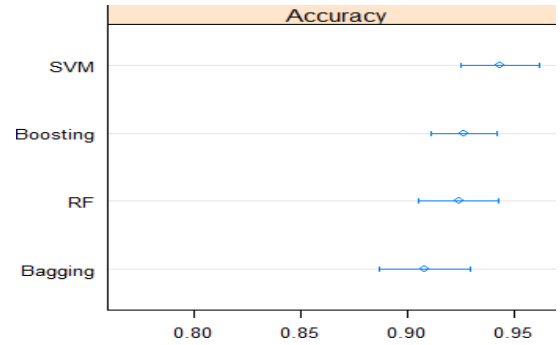

Figure 2.5 Accuracy of Pre-Processing Models on Training Data

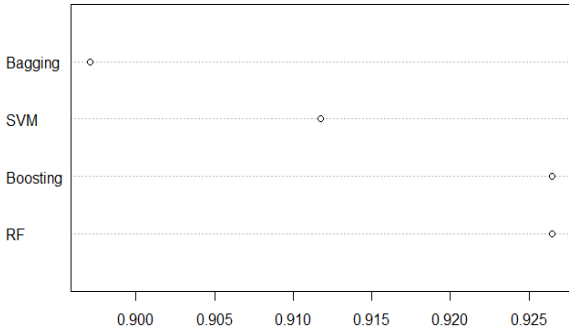

Figure 2.6 Accuracy of Pre-Processing Models on Validation Data.

Figure 2.5 is the Accuracy (correctly classified) of "Baseline Models with Pre-Processing" on Training Data. Figure 2.6 is the Accuracy of Figure 2.5 is the Accuracy (correctly classified) of "Baseline Models with Pre-Processing" on Validation Data. These two figures are to help us understand if there are any improvements after applying Pre-Processing to both Train and Validation Data.

3. <u>Tuning Parameters and Final Model.</u>

I will apply Pre-Processing (Standardize, PCA), and Tuning Parameters for Random Forest to improve the accuracy.

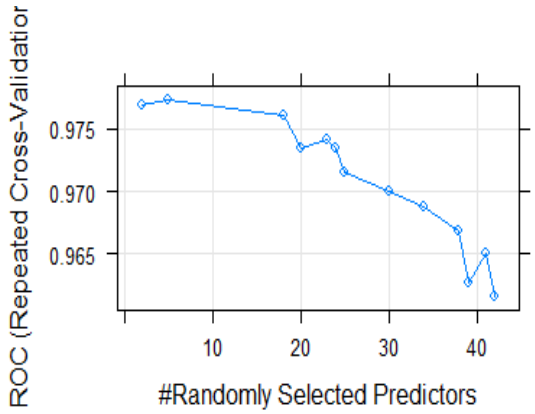

Figure 3.1 ROC vs. Randomly Selected Predictors using random search of Random Forest

mtry	ROC	Sens	Spec
2	0.9768946	0.9711111	0.8512821
5	0.9773219	0.9600000	0.9072650
18	0.9761610	0.9400000	0.9149573
20	0.9734900	0.9400000	0.9149573
23	0.9741026	0.9355556	0.9149573
24	0.9734900	0.9311111	0.9123932
25	0.9715171	0.9244444	0.9123932
30	0.9700712	0.9133333	0.9123932
34	0.9688533	0.9177778	0.9098291
38	0.9668020	0.9000000	0.9098291
39	0.9627707	0.9000000	0.9072650
41	0.9650142	0.8977778	0.9096154
42	0.9616311	0.9088889	0.9070513

Figure 3.2 The table of ROC with its mtry Predictors using random search of Random Forest.

4. <u>Performance Metrics.</u>

```
               Accuracy : 0.9459
                 95% CI : (0.8181, 0.9934)
    No Information Rate : 0.5405
    P-Value [Acc > NIR] : 6.688e-08

                  Kappa : 0.8912
 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9500
            Specificity : 0.9412
         Pos Pred Value : 0.9500
         Neg Pred Value : 0.9412
             Prevalence : 0.5405
         Detection Rate : 0.5135
   Detection Prevalence : 0.5405
      Balanced Accuracy : 0.9456
```

Figure 4.1 Confusion Matrix Test Data

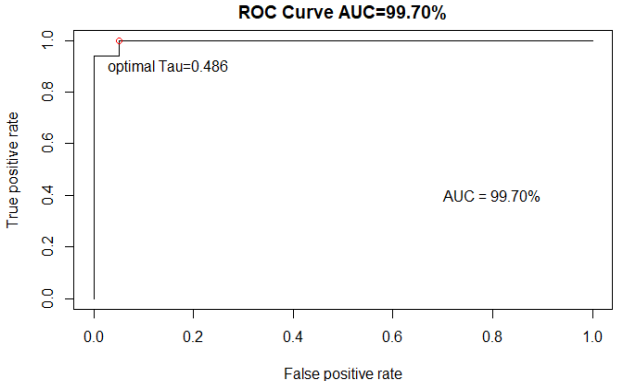

Figure 4.2 ROC with AUC and optimal threshold.

IV. DISCUSSION.

First of all, performing the Exploratory Data Analysis would help us to understand more about the dataset we are working with. There are 2 types of Data which are Numeric and Categorical in the dataset. Looking into the Pearson Correlation of continuous variables in the Training Data (*Figure 1.1*), there are highly correlated pairs of variables. If they are highly correlated, we should consider to remove some of them because they basically give the same info. However, I decided to take no action, and wait till next steps of Pre-Processing and Variable Selection. Looking at the structure of Categorical data (*Figure 1.2*), most of them are binary type and most of them have the name of "Criterion". This gives me some hints that few variables are not important and may give the same info.

In my Baselines models, I have used 4 different methods which are Random Forest, Bagging, Boosting, and SVM. Looking at the Accuracy metrics for Training Set (*Figure 2.1*), SVM performs the best among methods, Boosting and Random Forest have quite similar accuracy. However, Random Forest seems to have the lowest variance. The Accuracy for all 4 Baseline

Models significantly changes when we test with the Validation Test (*Figure 2.2*). Boosting and Random Forest have the same Accuracy and outperform SVM and Bagging. Comparing the Accuracy of Training Data and Validation Data, SVM and Bagging overfitting, Boosting is underfitting, and Random Forest is well fit. At this stage, I think Random Forest is chosen model because it gives the most consistent accuracy among 4 models and it is the "best" model in term of Bias-Variance Trade-Off.

I would like to check if my Pre-Processing and Variable Selection are able to improve the performance of my Baseline Models in term of Accuracy. I applied combination of "Center" and "Scale" to "Standardize" my data. This will make my data have Mean of zero and Standard Deviation of 1 and will benefit more when we apply PCA to the dataset. Applying PCA to dataset may benefit further steps of the project because PCA is multivariate technique to reduce data dimensions between highly correlated variables. In fact, We could successful express whole data in 3 less dimensions but still keeps 95% information (*Figure 2.4*). Moreover, doing the quick random forest with 500 trees, we can figure out the Variable Importance by measuring Mean Decrease Gini of each Variable (*Figure 2.3*). The Variables have low Mean Decrease Gini have direct proportion to their participation in growing trees. We can remove some least important variables such as Gender, Criterion32, Criterion20, Criterion10. By applying these 2 techniques to my data, I successfully reduced 7 columns in Dataset but still can expect almost full information.

The Accuracy of models after Pre-Processing and Variable Selection on training data (*figure 2.5*) are quite identical to Baseline Models, except the accuracy of Bagging is slightly improve. Similarly, The Accuracy of models after Pre-Processing and Variable Selection on Validation data (figure 2.6) is quite similar. However, The SVM is well fits and no longer overfitting as in Baseline Models. Random Forest still be the most consistent model in both Training and Validation data, in both Baseline and after transformation. I decided to apply Pre-Processing and Variable to Random Forest, to perform Tuning Parameters to improve the prediction ability of the model. This is also known as my Final Model. After Tuning Parameter, I would use my Final Model to predict for Test Data. Moreover, I would use the ROC metrics to tune parameters instead of Accuracy.

For Random Forest model, I would tune the "mtry" which is the random selected Predictors. In others words, my tuning parameter is the number of selected predictor at each split. From the result of tuning parameter for Random Forest, the best "mtry" for my model on Training Data is at 5 (*Figure 3.1*) because it has the highest ROC of 97.73% for Training Data (*Figure 3.2*).

Finally, I applied my "parameterized final model" to predict for Test Data and accessing the Performance Metrics by Confusion Matrix (*Figure 4.1*) and ROC curve (*Figure 4.2*). Looking at the Confusion Matrix (*Figure 4.1*), Accuracy is 94.59%, which means my Final Model correctly classified 35 observation out of 37 observations in Test Data without looking at Test Data in advanced. We have balanced Responses (Nociceptive, Neuropathic) so the Balanced Accuracy is quite high as expected which is 94.56%. The Sensitivity is 95.00%, which means 19 out 20 observations were correctly classified as Nociceptive. Similarly, The Specificity is 94.12%, which means 16 out of 17 observations from Neuropathic were correctly classified.

The ROC is a more in-depth look into the Sensitivity and Specificity *(Figure 4.2*). Particularly, the ROC of Binary Classification problem is the trade-off between Sensitivity and Specificity. In

order to measure ROC, we need to have the Predicted value as Probability. The ROC Curve for my "Parameterized Final Model" on Test Data is very close to 45-degree line, which means the AUC is extremely close to one. In fact, the Area Under Curve (AUC) for my ROC curve is 99.70%. Moreover, the optimal threshold for my ROC curve is at tau=0.486. At this threshold, the sum of Sensitivity and Specificity is maximized (Sensitivity = 100%, Specificity is 95%). In other words, at tau=0.486 my model is able to distinguish between Nociceptive and Neuropathic at its best.

V. CONCLUSION.

Comparing to limited classification abilities of Logistic Regression (~85% on both Training and Validation Data), Machine Learning algorithms are totally a better approach for this Binary Classification problem on this clinical dataset. There are some disadvantages with Machine Learning algorithms such as interpretability and computation efficiency but we clearly can optimize these 2 disadvantages. We can understand which variables are important in growing the trees in Tree-Based models by accessing the Variable Importance calculated by Mean Decrease Gini. This idea also support Variable Selection step in my project. Combining with Principle-Component Analysis, I could successfully reduce 7 variables in my data to improve the computation efficiency. Using Baseline Models to compare on both Train and Validation data, I decided to choose Random Forest (with Variable Selection, Pre-Processing) as my "best" model in term of Bias-Variance error. Furthermore, to improve my Final Model, I could tune parameters and figured out that with "mtry"=5 (random selected Predictors at each split) is the "best" parameter for my "Parameterized Final Model". In fact, my "Parameterized Final Model" performs really well on unseen data of Test Set. My model could correctly classified a whole Test Data with Accuracy of 94.59%, which is expected and quite similar to Accuracy on both Train and Validation data. For binary classification problem, Sensitivity, Specificity and trade-off of Sensitivity-Specificity (ROC curve) are critical performance metrics. ROC curve and its Area-Under-Curve (AUC) would provide how well my model can distinguish between Nociceptive and Neuropathic. My model has AUC of 99.70%, which is extreme strong ability to distinguish between 2 classes in this problem. Even more, at the threshold of tau=0.486, where sum of Sensitivity and Specificity is maximized, my "parameterized final model" can distinguish between Nociceptive and Neuropathic at its best with (Sensitivity = 100%, Specificity is 95%).

VI. REFERENCES.

@Manual{,
  title = {caret: Classification and Regression Training},
  author = {Max Kuhn. Contributions from Jed Wing and Steve Weston and Andre Williams and Chris Keefer and Allan Engelhardt and Tony Cooper and Zachary Mayer and Brenton Kenkel and the R Core Team and Michael Benesty and Reynald Lescarbeau and Andrew Ziem and Luca Scrucca and Yuan Tang and Can Candan and Tyler Hunt.},
  year = {2018},
  note = {R package version 6.0-80},
  url = {https://CRAN.R-project.org/package=caret},
}

@Manual{corrplot2017,
  title = {R package "corrplot": Visualization of a Correlation Matrix},
  author = {Taiyun Wei and Viliam Simko},
  year = {2017},
  note = {(Version 0.84)},
  url = {https://github.com/taiyun/corrplot},
}

@Article{,
  entry = {article},
  title = {ROCR: visualizing classifier performance in R},
  author = {T. Sing and O. Sander and N. Beerenwinkel and T. Lengauer},
  year = {2005},
  journal = {Bioinformatics},
  volume = {21},
  number = {20},
  pages = {7881},
  url = {http://rocr.bioinf.mpi-sb.mpg.de},
}