ABSTRACT

Lower back pain can be caused by a variety of problems with any parts of the complex, interconnected network of spinal muscles, nerves, bones, discs or tendons in the lumbar spine. Here, it is focused to classify subjects according to the type of lower back pain they experience, using a collection of 380 subjects and 32 clinical indicators (12 numerical, 11 ordinal and 19 categorical). So, here each subject is classified as **nociceptive** (pain that's likely to be caused by the body being harmed which includes mechanical or physical damage to non-neural tissue) or **neuropathic** (Pain initiated or caused by a primary lesion or dysfunction in the nervous system). To get best accurate classification we are using range of algorithms like Logistic Regression, Classification Trees, Support Vector Machine etc. The methods like bootstrapping, bagging and boosting will also be done to increase the efficiency of the classifier. The merits and demerits of each algorithm will be discussed along with the results after running these algorithms on our dataset. The model with higher accuracy for most of the cases will be selected as the best model and will be used to classify the type of lower back pain they experience.

CONTENTS

1 INTRODUCTION

Low back pain is one of the leading cause of disability in the modern world and can be caused due to number of factors, especially genetic or environmental factors. The problems may be with any parts of the complex, interconnected network of spinal muscles, nerves, bones, discs or tendons in the lumbar spine.

The aim of this analysis is to classify the pain type to nonciceptive and neuropathic by analysing the diagnosis information with the best classifier selected from a range of classifier. The performance of each classifier also depends on what dataset being used. Some, may perform well on the train data but show low accuracy with the new data. So, it is always good to find out which classifier classifies the observations more accurately. The classifiers used here in this analysis are:

1. Logistic Regression
2. Support Vector Machine
3. Classification Trees
4. Random Forest
5. Bagging
6. Boosting

Also, to improve the efficiency of the classifiers the methods like bootstrapping will also be performed. The complete analysis will be done using R statistical software.

1.1 Dataset

The “backpain” dataset contains diagnosis information regarding subjects suffering from lower back pain. There are 380 observations and 32 variables (features) in the dataset. Among the 32 variables 12 are numerical, 1 is ordinal and the rest 19 are categorical. The response categories provide a clinically meaningful binary classification of lower back pain. The two categories are **nonciceptive** (pain arising from actual damage to non-neural tissue with a normal functioning nervous system) and **neuropathic** (pain caused by primary lesion or dysfunction in the nervous system).

1.1.1 Details of variables

• PainDiagnosis :- Expert pain diagnosis (as reference standard) • Age :- Age of the patient. • Gender :- Gender of the patient • DurationCurrent :- Duration of current pain. • PainLocation :- Pain location. • SurityRating :- Surity rating of expert clinical diagnosis.	• Criterion6 :- More constant, unremitting. • Criterion7 :- Burning, shooting, sharp, electric-shock like. • Criterion8 :- Localised to area of injury, dysfunction. • Criterion9 :- Referred in dermatomal, cutaneous distribution.

• RMDQ :- Roland Morris Disability Questionnaire score. • vNRS :- Verbal NRS for pain intensity. • SF36PCS :- SF36 Physical Component Summary score. • SF36MCS :- SF36 Mental Component Summary score. • PF :- Physical Functioning • BP :- Bodily Pain. • GH :- General Health. • VT :- Vitality. • MH :- Mental Health. • HADSAnx :- HADS Anxiety score. • HADSDep :- HADS Depression score. • Criterion2 :- Pain assoc'd trauma, pathology, movt. • Criterion4 :- Pain disproportionate to injury, pathology.	• Criterion10 :- Widespread, non-anatomical distribution. • Criterion13 :- Disproportionate, non-mechanical pattern to aggs + eases. • Criterion19 :- Night pain, disturbed sleep. • Criterion20 :- Responsive to simple analgesia, NSAIDS. • Criterion26 :- Pain with high levels of functional disability. • Criterion28 :- Consistent, proportionate pain reproduction on mechanical testing. • Criterion32 :- Localised pain on palpation. • Criterion33 :- Diffuse, non-anatomic areas of pain on palpation. • Criterion36 :- Positive findings of hyperpathia

2 METHODS

2.1 Data Wrangling

Before applying data to the classifiers it is good to check whether data has any impurities. For that the summary statistics of the data is made. Analysing the summary statistics we can see that there are no missing values in the data. Also there is sufficient amount of data in each category in the response variable. We need to ensure that our response variable is coded as a factor variable. The ordinal variable 'SurityRating' is coded as numerical, so it's changed to categorical. The test, train and validation split will be done after bootstrapping.

2.2 Classifier Algorithms Under Test

The following classifier models are used in this analysis for predicting the type of the back pain;

2.2.1 Logistic Regression

Logistic regression is one of the most popular classification approach when the response variable is dichotomous (binary). Unlike choosing parameters that reduces sum of squares errors, logistic regression chooses parameters that maximize the likelihood of observing the sample values.

For creating a logistic regression model we have used "multinom" function in "nnet" library. In this the log odds of the predicted outcomes are modelled as linear combination of predictor variables.

2.2.2 Support Vector Machine

The support vector machine tries to find out the best hyperplane to separate to classes. The SVM will get a great performance boost if it is combined with kernels. Because these kernels will project the data which are not linearly separable to higher dimensions where we can easily separate them. Which means we can fit a linear classifier to a non-linear relationship.

To generate SVM model, we have used "ksvm" method in "kernlab" library which is a kernel based SVM. The default radial basis kernel (Gaussian) with default tuning parameter is used here.

2.2.3 Classification Tree

Classification trees are based on the splits of categorical variables, ordinal and even numerical variables (split will be based on some thresholds), or a mix of all types of predictor variables. Split choices can be based on the entropy or the Gini index (impurities). The tree is created from the root (top) node and grown downwards. And the root node will be the best predictor. Here, the "rpart" library will be used to generate the classification tree model.

2.2.4 Random Forest

Random forest is basically an ensemble method, it creates multiple decision trees and merges them together to produce accurate and stable results. That is, it adds more randomness while growing the trees. It only takes the random subset of the predictor variables for splitting a node. Another feature of random forest is that; we can check the variable importance of the predictor variables. Here, the "randomForest" library will be used to generate the random forest model.

2.2.5 Bagging

Bagging also known as bootstrap aggregation is one of the popular ensemble method which uses results from multiple prediction algorithms. Bagging can be applied to methods which have high variance like the decision trees. Bagging will also help in reducing overfitting of the data. The bagging is done using "bagging" function in "adaboost" library.

2.2.6 Boosting

Boosting is also an ensemble method for improving model prediction accuracy which can be applied to any of the existing classification algorithm. The main idea behind boosting is that they analyse the performance of the model and then concentrate more on the misclassified observations. By increasing the importance of the misclassified observations and refitting the model the performance of the model can be increased. But boosting is sensitive to the label noise or outliers. The boosting is done using "boosting" function in "adaboost" library.

To improve the efficiency of the classifiers **bootstrapping** is done on the entire given dataset. Bootstrapping is a sampling technique which produces samples of dataset given replacement. Bootstrapping improves the prediction accuracy of data which is not present on the train data, which means accuracy increases when a new data is given. Bootstrapping can be applied to every classifier to improve its efficiency.

2.3 ALGORITHM FOR FINDING BEST MODEL

The algorithm used for finding the best model is given in the block diagram Fig 1 below. These algorithms will be running for 100 times, on each iteration the samples of data for test, train and validation will be different which will increase the efficiency of the algorithm. The train data will be used to fit the model, the validation data will be used to test accuracy of each model which will be used to select best model in each iteration and test will be used to predict with the best model selected.

Algorithm steps for finding best model (Block Diagram Fig 1):

Step 1 – Bootstrapping and data split

Bootstrap sampling on the data is done on the entire dataset given with the number samples equal to the number of observations. Approximately 0.6% of the data in the dataset will be there in the resulting bootstrap sample which is our train data. The data which is omitted during bootstrapping is split in to validation (50%) and test (50%).

Step 2 – Generating models with train data

The above mentioned (Section 2.2) algorithms will used to create model with the test data

Step 3 – Predicting with validation data and taking the best model

The validation data is used to find the accuracy of each fitted models. And the model with best accuracy is chosen as the best model in that iteration and it is logged. The test data is given to check the performance of the best model with the test data.

Step 4 – Step 1 – 4 is repeated for 100 iterations

Step 5 – Analysing test statistics and best model count

The model, which came most frequently as the best model in each iteration is taken as our best model.

Block Diagram of Algorithm Used

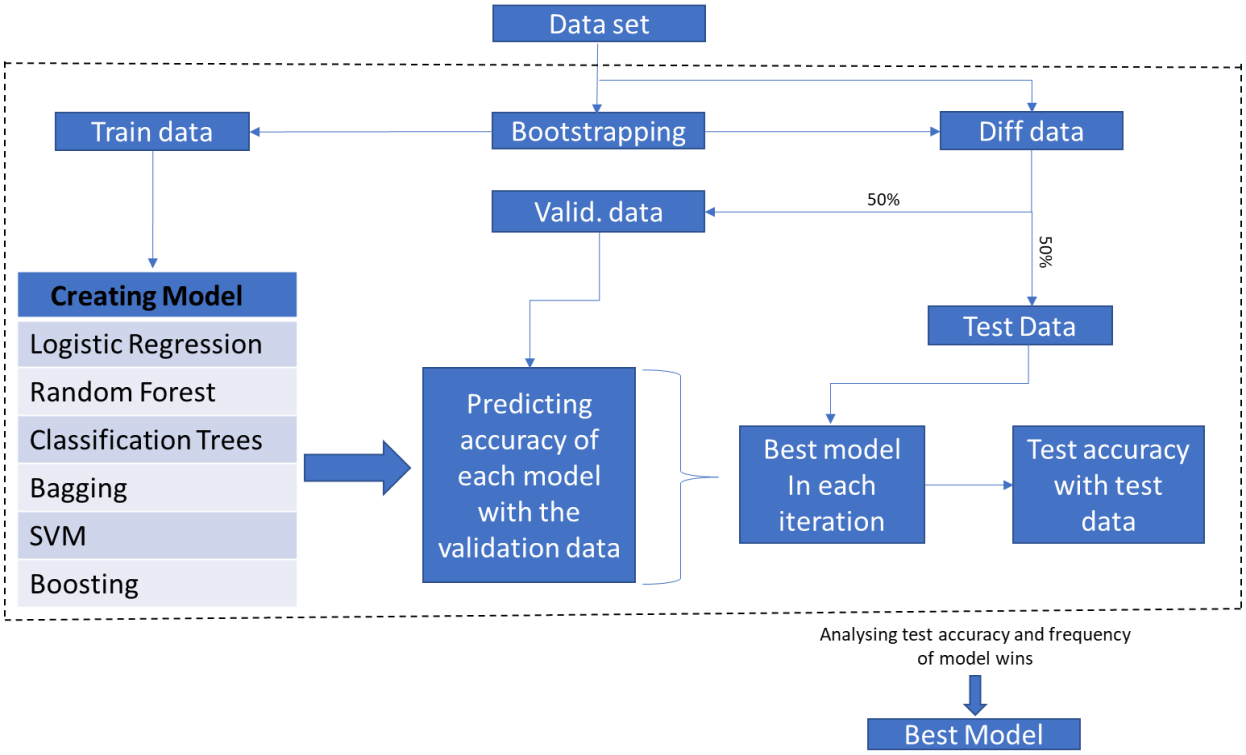


Fig: 14

3 RESULTS

3.1 Models performance on validation data

3.1.1 Accuracy table

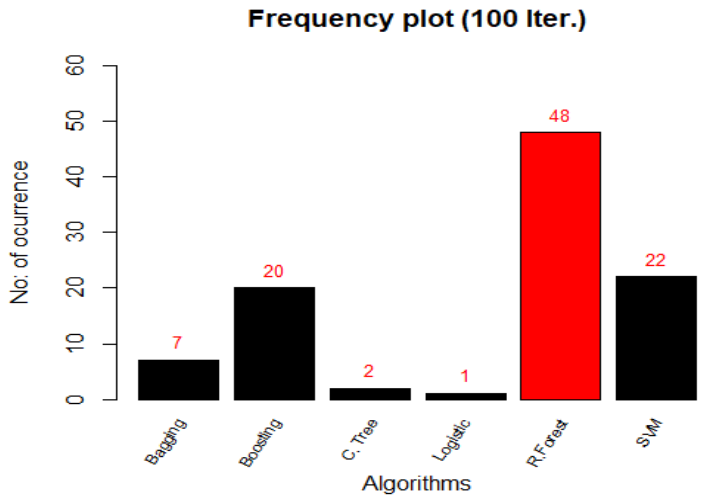
Statistic	Logistic	R.Forest	SVM	C. Tree	Bagging	Boosting
Min.	0.7941	0.8657	0.8382	0.7391	0.7632	0.8649
1st Qu.	0.8453	0.9087	0.8996	0.854	0.8841	0.9014
Median	0.8723	0.9275	0.9178	0.8788	0.9034	0.9275
Mean	0.8695	0.9268	0.9193	0.8731	0.8997	0.9239
3rd Qu.	0.8971	0.9446	0.9412	0.8989	0.9242	0.9439
Max.	0.9429	0.9863	0.9846	0.9577	0.9861	0.9855

Table 1

3.1.2 Models frequency statistics

Algorithm	Wins
Bagging	7
Boosting	20
Classification Tree	2
Logistic	1
Random Forest	48
SVM	22

Table 2



3.2 Best Model (each iteration) performance on test data

3.2.1 Accuracy table

Algorithm	Frequency	Max	Mean	Min
Bagging	7	0.955882	0.914958	0.883117
Boosting	20	0.986301	0.929293	0.891892
Classification Tree	2	0.924242	0.906566	0.888889
Logistic	1	0.942857	0.942857	0.942857
Random Forest	48	1	0.924105	0.864865
SVM	22	0.955882	0.876388	0.785714

Table 3

3.3 Variable importance (Random forest model)

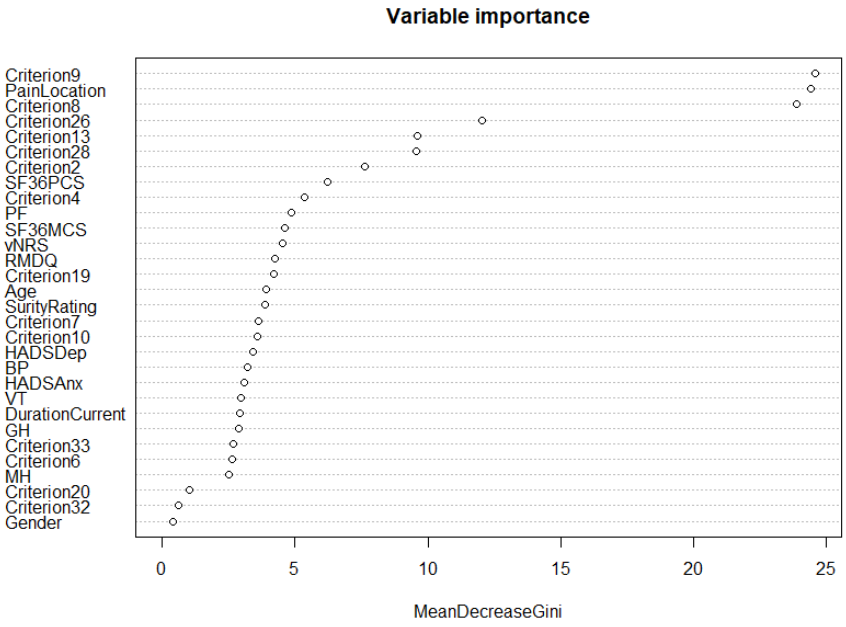


Fig 3

4 DISSCUSION

From **Table 1** we can see the accuracy statistics of the models generated when the new validation data is given. All algorithm has showed a decent performance on the data. Apart from classification tree and logistic regression the max accuracy was 98%. Lowest accuracy 73% was given by the classification tree followed by bagging 76% and logistic regression 79%. The most important statistical measure here will be the mean because it will give the average accuracy from the 100 iterations. The Random forest has the highest mean accuracy of 92.6% followed by boosting and SVM.

The **Table 2** and **Fig 2** depicts the count of best model in each iteration. Checking these we can see that the Random forest was chosen as best model for 48 times which was followed by SVM for 22 times and boosting for 20 times. The classification tree and logistic regression appeared only for 2 and 1 times respectively.

Table 3 shows the performance of the models when test data is given. Random forest has shown maximum accuracy of 100% which is followed by boosting (98%) and SVM (98%). The statistics of classification tree, logistic regression and bagging doesn't make much sense since it is only very few in count. The minimum accuracy shown by Random forest, boosting and SVM was 86%, 89% and 78% respectively. Checking the mean accuracy for the models is not very appropriate since all have different number of occurrence even though the random forest with 48 occurrences has given a mean accuracy of 92%. The SVM and boosting has a mean accuracy of 87% and 92% respectively

Fig 3 gives the variable importance plot, which indicates the important variables which are contributing to the random forest model. The x axis shows the mean decrease in gini which is the average of decrease in the node impurity. So, a higher value of mean decrease in gini values indicates the predictor variable highly contribute to the response variable. If the mean decrease in gini value is less, then the predictor variable is less important. So, from the plot the most important variable is at the top which is 'Criterion9' and the importance of the variable decreases as move down. So, the lowest important variable will be Gender.

5 CONCLUSION

From the discussions above it's clear that Random forest has shown great performance on test and validation data. Also, it was the most frequent best model in each iteration. So, we can choose **Random Forest** as our best model in this analysis. It was followed by boosting, even though the model frequency is slightly less than SVM, the average accuracy of the model with test and validation data is better. The logistic regression and classification trees shown less performance comparing to others which may be because of the over fitting issues in the classification trees and logistic regression.

One of the great quality in random forest algorithm is that it is very easy to find the importance of each predictor variables to the response variable. From the discussion on variable importance from above, the top 3 factors which have greater important on deciding the type of lower back pain suffered are;

1) Criterion 9 – "Referred in dermatomal, cutaneous distribution."
2) PainLocation
3) Criterion 8 - "Localised to area of injury, dysfunction."

Also factors like gender, Criterion32 – "Localised pain on palpation.", Criterion20 - "Responsive to simple analgesia, NSAIDS." etc are not much important on deciding the type of lower back pain.