"Spot checking of K-NN, Naïve Bayes, and Logistic regression algorithm for diabetes dataset in R"

ALY6020 Final Project for Predictive Analytics

Submitted to:

Dr. Michael Prokle
College of Professional Studies
Northeastern University, Boston

Submitted by:

Anupam Maheshwari
Academic Term: Summer Quarter 2018
Northeastern University, MA

Abstract

The Paper talk in brief about what is spot checking and how to implement it. The description of each tested algorithm (K-NN, Naïve Bayes, and logistic regression) is made along with its classification analysis over the diabetes dataset. The paper covers classification algorithm and doesn't include any regression algorithm. The united result of accuracy of each of the algorithm is plotted and compared and the best model for classification of diabetes dataset is selected. The paper includes strength, weakness and accuracy calculation of each algorithm.

Introduction

With the evolution of machine learning algorithm and it's wide end integration in various industry sector, their utility shot up high. Every industry now wants to implement machine learning algorithm to predict data or to fetch out insights using them.

Due to this evolution a new research field pop up which deals with what kind of algorithm to implement on a dataset. Each dataset is unique and comes with its own problem statement. The various methodology used for implementing algorithm for a dataset are as follows:

- Using legacy experience – This practice includes using past experience for deciding which algorithm to implement the algorithm. This kind of implementation gets stuck if the algorithm is new.
- Trial and Error – This kind of methodology suffers major setback as there are N-number of machine learning algorithm which can be used on same dataset. This is not feasible for real world business scenario what the data throughput and insights are needed in real time.
- Spot checking – Spot checking is a process in which the mixture of different kind of machine learning algorithm are tested for a same dataset to check which is the algorithm that predicts and classify the set most accurately. The spot checking is done for both classification and regression problems.

Today in the industry lot of people are inclined to use spot checking process as it adds performance value and optimization in the machine learning process of dataset, and help in making calculated choice for selection of algorithm based on the accuracy level.

Objective

The objective of this paper stands as for selecting the best available algorithm among K-NN, Naïve Bayes, and Logistic Regression for a Indian Pima Indians Diabetes Database which is available in R package library.

The major task is analysing the accuracy level of each of the algorithm based on learning, classification and runtime. "Note: Since the database is small the runtime difference can be calculated."

Dataset

I have taken Pima Indians Diabetes dataset for the analysis and comparing different classification methods. The data frame consists of 768 observations performed on 9 variables.

The dataset is available in MLbench package.

Data format :

1 Number of times pregnant

2 Plasma glucose concentration (glucose tolerance test)

3 Diastolic blood pressure (mm Hg)

4 Triceps skin fold thickness (mm)

5 2-Hour serum insulin (mu U/ml)

6 Body mass index (weight in kg/(height in m)^2)

7 Diabetes pedigree function

8 Age (years)

9 Class variable (test for diabetes)

Original owners: National Institute of Diabetes and Digestive and Kidney Diseases

```
> head(dataset)
  pregnant glucose pressure triceps insulin mass pedigree age diabetes
1        6     148       72      35       0 33.6    0.627  50      pos
2        1      85       66      29       0 26.6    0.351  31      neg
3        8     183       64       0       0 23.3    0.672  32      pos
4        1      89       66      23      94 28.1    0.167  21      neg
5        0     137       40      35     168 43.1    2.288  33      pos
6        5     116       74       0       0 25.6    0.201  30      neg
>
```

Figure 1: Data Preview

Selecting algorithm for dataset

The basic criteria followed while selecting the pool of testing algorithm for any dataset is about choosing the combination of linear and non-linear algorithms.

Linear Algorithm- These algorithm make hypothesis on the basis of how the function is modelled. These are methods that make large assumptions about the form of the function being modelled. These algorithm comes with high bias rates but on the other hand are quick to plot results. These algorithm does get priority to get tested first as they are quick learner and takes less implementation time.

Non-Linear Algorithm: These algorithm make few hypothesis about the data and usually implement either predefined or the guided user defined function. This results in higher accuracy with increased dependency on variance. They usually take large memories and run time for execution.

Note: The algorithm used in this paper are :

- Linear methods: Logistic Regression.
- Non-Linear methods: K-NN and Naive Bayes

Algorithm Selected for Testing

Naive Bayes (NB)

This is a classifying algorithm which uses data about prior events to estimate the probability of future events. Typically it's best applied to problems in which the information from numerous independent/dependent attributes should be considered simultaneously in order to estimate the probability of an outcome. While many algorithm typically ignore features with weak effects, this technique uses all available features for prediction as it states 'the combination of group of weak effect feature can create a major prediction change'.

Naïve Bayes has very unique Laplace Estimator for the classification of spam text combination even if the combination is not recorded before. I believe this feature is one of the strongest add on for the real time scenarios.

k-Nearest neighbours

K- nearest neighbour classification perform direct classification of data without building model on the first hand. With no model creation the only calculatable and adjustable parameter is the value of K in the algorithm which is NN in the estimate of class membership. Changing the value of *k*, the model can be made more or less flexible. Note: Value of K can't be a negative integer.

Logistic Regression

"It is used for predicting binary result (1 / 0, Yes / No, True / False) on given a set of independent variables. To represent binary / categorical outcome, we use dummy variables. You can also think of logistic regression as a special case of linear regression when the outcome variable is categorical, where we are using log of odds as dependent variable. In simple words, it predicts the probability of occurrence of an event by fitting data to a logit function." [2]

Methodology

Test options refers to the technique used to evaluate the accuracy of a model on unseen data. They are often referred to as resampling methods in statistics. As all three algorithm perform and work on individual basis the paper evaluate them based on the following methodology.

Step 1 is about Loading required libraries and dataset in the global environment of R.

```
> library(mlbench)
> library(caret)
>
> # load data
> data(PimaIndiansDiabetes)
> # rename dataset to keep code below generic
> dataset <- PimaIndiansDiabetes
>
```

Figure 2: Loading data and libraries

Step 2 includes pre-processing the data. It has been observed that lot of algorithm reflects improved accuracy and runtime when data is optimized and cleaned. To check each algorithms accurate performance power the train function of caret is used on the dataset. The caret function put all of the variable on the same scale.

Note: data pre-processing is performed just before the training of the dataset.

```
> #data preprocessing
> control <- trainControl(method="repeatedcv", number=13, repeats=3)
> seed <- 5
> metric<-"Accuracy"
> preProcess=c("center", "scale")
```

Figure 3: Data pre-processing

Step 3 involves three listed models trained with the Pima Indians Diabetes dataset, and validated for accuracy against the cross-validation data set.

"Cross Validation of 5 folds or 10 folds provide a commonly used trade-off of speed of compute time and generalize error estimate. Here the function is implemented for 10 fold with 3 times repeat validation." [3]

Note: Assigning a random number seed to a variable re-set the random number generator before training each model algorithm . It ensures that each algorithm is evaluated on exactly the same splits of data and assisting in comparisons on same line

Performance Parameters :

- **Accuracy**- This is equivalent to total number of accurately predicted value in the dataset.
- **Kappa**- Accuracy that takes the base distribution of classes into account.

Training and testing for K-NN

```
> #kNN model
> set.seed(seed)
> fit.knn <- train(diabetes~., data=dataset, method="knn", metric=metric, preProc=c("center", "scale"), trControl=control)
> fit.knn
k-Nearest Neighbors

768 samples
  8 predictor
  2 classes: 'neg', 'pos'

Pre-processing: centered (8), scaled (8)
Resampling: Cross-Validated (13 fold, repeated 3 times)
Summary of sample sizes: 708, 709, 708, 710, 710, 709, ...
Resampling results across tuning parameters:

  k  Accuracy   Kappa
  5  0.7434127  0.4118895
  7  0.7416373  0.4087593
  9  0.7412554  0.4023362

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 5.
```

Figure 4: Testing K-NN model and printing object result

Training and testing for Naïve Bayes

```
> fit.nb
Naive Bayes

768 samples
  8 predictor
  2 classes: 'neg', 'pos'

No pre-processing
Resampling: Cross-Validated (13 fold, repeated 3 times)
Summary of sample sizes: 708, 709, 708, 710, 710, 709, ...
Resampling results across tuning parameters:

  usekernel  Accuracy   Kappa
  FALSE      0.7565709  0.4514667
   TRUE      0.7530646  0.4368514

Tuning parameter 'fL' was held constant at a value of 0
Tuning parameter 'adjust' was held constant at a value of 1
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were fL = 0, usekernel = FALSE and adjust = 1.
>
```

Figure 5: Testing Naive Bayes model and printing object result

Training and testing for Logistic Regression

```
> # Logistic Regression model
> set.seed(seed)
> fit.glm <- train(diabetes~., data=dataset, method="glm", metric=metric, trControl=control)
> fit.glm
Generalized Linear Model

768 samples
  8 predictor
  2 classes: 'neg', 'pos'

No pre-processing
Resampling: Cross-Validated (13 fold, repeated 3 times)
Summary of sample sizes: 708, 709, 708, 710, 710, 709, ...
Resampling results:

  Accuracy   Kappa
  0.7760199  0.4791001
```

Figure 6: Testing Logistic Regression model and printing object result

Conclusion and Insights

After training dataset with each of the algorithm model, the object of each model is resampled for creating summary table. The table there reflects the minimum, maximum, median, and mean of the accuracy of all three models.

The result clearly shows the comparison between the listed three models and the following can be quickly inferred:

- Logistic Regression is the best classification method for the given Diabetes dataset with 77.6% accuracy.
- While K-NN and Naïve Bayes have marginal difference in accuracy as 74.34% and 75.65% respectively.
- Logistic Regression outperform with kappa indicator standing at 0.47
- Here Naïve Bayes outperform K-NN with kappa indicator standing at 0.45 and 0.41 respectively.

```
> #listing all the model object together
> results <- resamples(list( logistic=fit.glm,
+                            knn=fit.knn, nb=fit.nb))
>
> # Table comparison
> summary(results)

Call:
summary.resamples(object = results)

Models: logistic, knn, nb
Number of resamples: 39

Accuracy
             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
logistic 0.6500000 0.7543103 0.7796610 0.7760199 0.8119521 0.8500000    0
knn      0.6500000 0.7142655 0.7457627 0.7434127 0.7712644 0.8305085    0
nb       0.6333333 0.7288136 0.7500000 0.7565709 0.7899718 0.8644068    0

Kappa
             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
logistic 0.1860465 0.4243987 0.4824561 0.4791001 0.5648022 0.6666667    0
knn      0.2045455 0.3531403 0.4186047 0.4118895 0.4741914 0.6180556    0
nb       0.1760300 0.3925578 0.4680851 0.4514667 0.5353261 0.6978233    0
```

Figure 7: Accuracy result of individual model for Diabetes dataset

Graphical Comparison of Performance Matrix

The figure 7 shows the relative comparison between K-NN, Naïve Bayes, and Logistic Regression base on accuracy and kappa indicators.

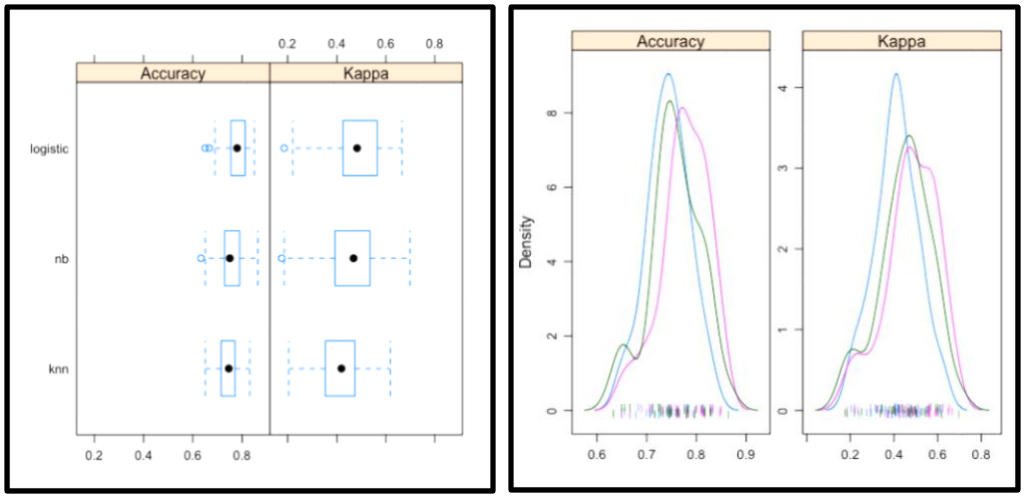

Figure 8: Boxplot and Density Plot comparison of accuracy for Logistic Regression, K-NN, Naive Bayes

Result

After performing and testing for K-NN, Naïve Bayes, and Logistic Regression for diabetes dataset I can clearly see and choose Logistic regression for data classification.

Future study:

Due to restricted size of database and limitation with the system hardware the test can't be evaluated on base of hardware. I am looking forward to work on following problem statements as the runtime, parallel processing and hardware also plays important part for machine learning algorithm. The major investigation stands for

- How different machine affect algorithm runtime?
- How parallel processing and math kernel library affects the efficiency of algorithm since R spans single threaded process?

References and Citations

1. Predictive Analytics using R: Jeffrey Strickland, Retrieved 12th August, 2018
2. https://www.analyticsvidhya.com/blog/2015/11/beginners-guide-on-logistic-regression-in-r/https://machinelearningmastery.com/evaluate-machine-learning-algorithms-with-r/
3. https://machinelearningmastery.com/spot-check-machine-learning-algorithms-in-r/ - Testing and Methodology
4. https://rpubs.com/omicsdata/160333 -Graphical Comparison of Performance Matrix
5. http://ugrad.stat.ubc.ca/R/library/mlbench/html/PimaIndiansDiabetes.html - Dataset Description
6. Assignment- 1 Submitted for Predictive Analytics week-1
7. Assignment- 2 Submitted for Predictive Analytics week-2
8. Assignment- 5 Submitted for Predictive Analysis week-5