CROSS VALIDATION AND MODEL SELECTION IN R

Contents

[Executive Summary 2](#_Toc468234684)

[Introduction 2](#_Toc468234685)

[Cross Validation 3](#_Toc468234686)

[Types of Cross validation techniques 3](#_Toc468234687)

[Holdout Method 3](#_Toc468234688)

[K-fold cross validation 4](#_Toc468234689)

[Leave-one-out cross validation 5](#_Toc468234690)

[Problem Statement 7](#_Toc468234691)

[Solution 7](#_Toc468234692)

[Confusion Matrix for XGBoost Model on validation set approach 8](#_Toc468234693)

[Metrics 8](#_Toc468234694)

[The validation set approach 9](#_Toc468234695)

[Cross-Validation approach 9](#_Toc468234696)

[Results of 5-fold cross validation for logistic regression 11](#_Toc468234697)

[Results of 5-fold cross validation for XGBoost 11](#_Toc468234698)

[Results of 5 fold cross validation for random forest 11](#_Toc468234699)

[Confusion matrix for random forest for cross validation approach 12](#_Toc468234700)

[Model Ensembling 12](#_Toc468234701)

[Confusion Matrix for ensemble model using majority voting 14](#_Toc468234702)

[Other applications of cross validation 14](#_Toc468234703)

[References 15](#_Toc468234704)

# Executive Summary

This report showcases the importance of cross validation techniques and model Ensembling. For our analysis, we took a bank marketing data set from UCI repository having 690 observations and 15 dimensions. This is a classification problem with outcome as a binary variable. We followed approaches of validation techniques to demonstrate how the model accuracy improves over one another along with the model ensembling technique using majority voting.

Firstly, we performed the basic validation technique i.e. simple validation hold out method and scored the results on the validation data using four models random forest, xgBoost, Decision tree and logistic regression. The highest accuracy is observed on the xgboost model which is around 85%. We got an accuracy of 79% when scored results on the test data.

Secondly, we performed the k fold cross validation technique to evaluate the models. For 5 fold cross-validation approach, the log loss values are not consistent for logistic regression, hence we did not choose this model. The xgboost and random forest log loss values are consistent ,but the values of xgboost are high when compared to the random forest. Hence , we choose random forest for the 5 fold cross validation technique and scored the test results where the accuracy improved to 87%.

Lastly, the third technique we used to evaluate the model performance is model ensembling using majority voting. It is a technique used to combine two or more models for improving model accuracy. We combined three models decision tree, random forest and xgboost by taking into account the maximum number of votes given by all the models and scored the test results which improved the accuracy to 88.33%.Hence,these techniques are very efficient in evaluation and selection of models which gives accurate results on the real world data as the model selection is more robust.

# Introduction

The most essential problem in model fitting is to fit a model with optimum bias-variance. There are several techniques used to obtain bias-variance trade-off. If the data is not rich and we have limited number of observations, we cannot afford to use data for validation as we will be using less data for training our model. Cross-Validation is one of the important techniques can be used in this kind of scenario. K-fold Cross validation is being used in this paper to show the power of cross-validation to improve model accuracy. Also, one of the important techniques to improve model accuracy is Ensemble models. Using Voting technique in ensemble models, the accuracy of the models can be improved. In this paper we have shown how to improve the accuracy of a model using voting technique. Also, there are few limitations of ensemble modelling using voting, which are discussed in this paper in detail.

# Cross Validation

Cross validation is the statistical technique used to validate the prediction power of different models on a given dataset. With this technique, we can assess how the results of a [statistical](https://en.wikipedia.org/wiki/Statistics) analysis will generalize to an independent data set. Using cross validation, we reduce the bias involved in our models and aim towards building a robust model which can do well on future data.

## **Types of validation techniques**

### Holdout Method

The **holdout method** is the simplest kind of validation. The data set is separated into two sets, called the training set and the validation set. The model is trained by using the training set only. Then the model obtained is used to predict the output values for the data in the valdation set (it has never seen these output values before). The accuracy/errors on the validation set are observed to select the model. The advantage of this method is that it is usually preferable to the residual method and takes no longer to compute. However, its evaluation can have a high variance. The evaluation may depend heavily on which data points end up in the training set and which end up in the validation set, and thus the evaluation may be significantly different depending on how the division is made.

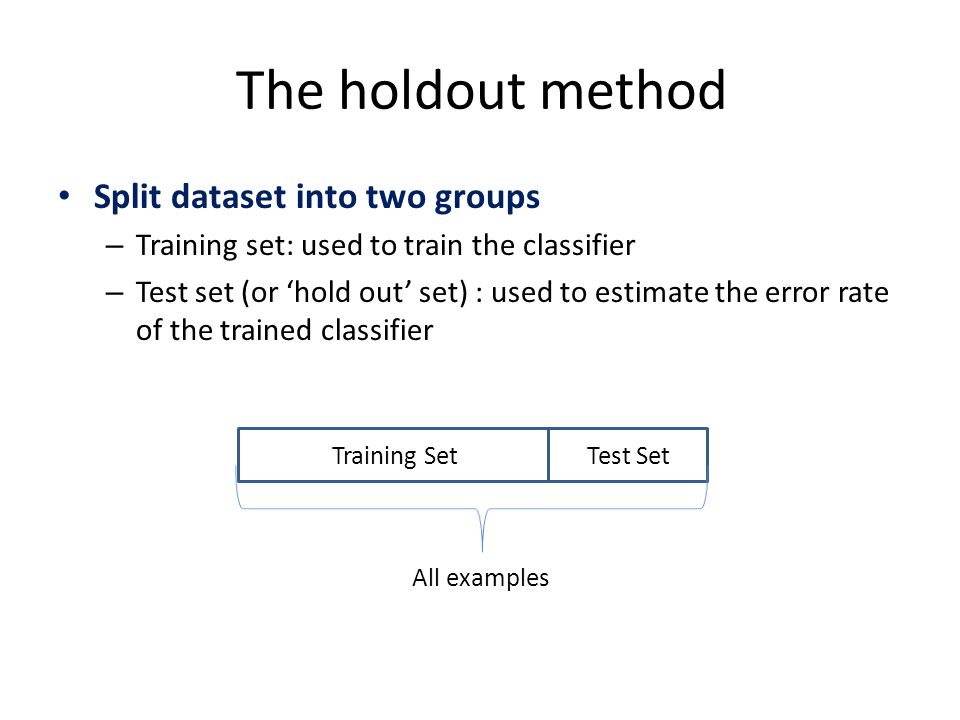


Figure 1.1 Hold out method [1]

### K-fold cross validation

**K-fold cross validation** is one way to improve over the holdout method. The data set is divided into k subsets, and the holdout method is repeated k times. Each time, one of the k subsets is used as the test set and the other k-1 subsets are put together to form a training set. Then the average error across all k trials is computed. The advantage of this method is that it matters less how the data gets divided. Every data point gets to be in a test set exactly once, and gets to be in a training set k-1 times. The variance of the resulting estimate is reduced as K is increased. The disadvantage of this method is that the training algorithm must be rerun from scratch k times, which means it takes k times as much computation to make an evaluation. A variant of this method is to randomly divide the data into a test and training set k different times. The advantage of doing this is that you can independently choose how large each test set is and how many trials you average over.

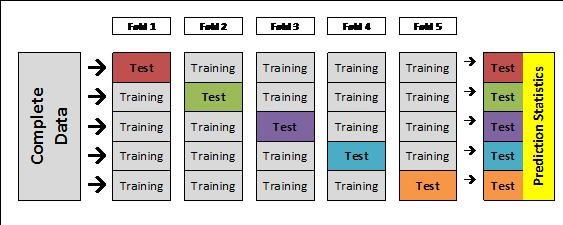


Figure 1.2 K fold cross validation[2]

### Leave-one-out cross validation

**Leave-one-out cross validation** is K-fold cross validation taken to its logical extreme, with K equal to N, the number of data points in the set. That means that N separate times, the function approximator is trained on all the data except for one point and a prediction is made for that point. As before the average error is computed and used to evaluate the model. The evaluation given by leave-one-out cross validation error (LOO-XVE) is good, but at first pass it seems very expensive to compute. Fortunately, locally weighted learners can make LOO predictions just as easily as they make regular predictions. That means computing the LOO-XVE takes no more time than computing the residual error and it is a much better way to evaluate models.

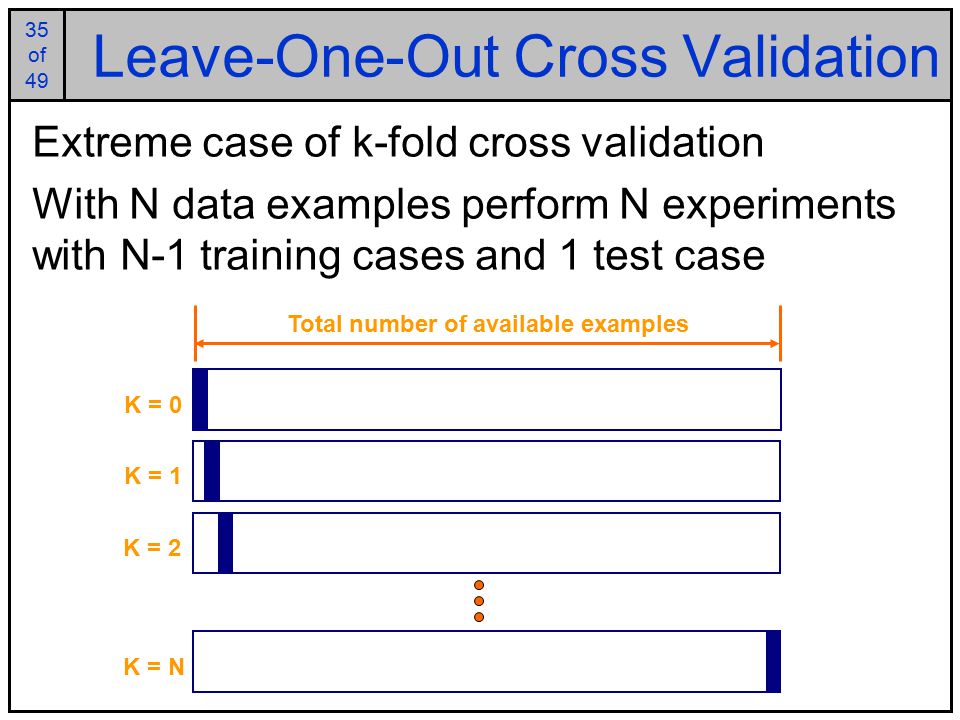


Figure 1.3 Leave one out cross validation[3]

In this report, we will first show the results from simple holdout method and then compare its results with the k fold cross validation method.

# Problem Statement

On a given dataset we can build multiple models having varied accuracy but, as new data comes in we must be sure that the previously fitted model on our existing dataset will work on it. We should come up with a model that is robust enough to generalize and handle new data and at the same time gives better accuracy.

# Solution

For model simplicity and equal comparison of models we have taken all the independent variables in the models that we have built for both holdout sampling and k fold cross validation.

We have created 4 different models:

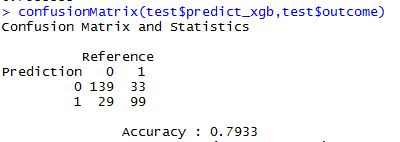
* *Logistic Regression*
* *Random Forest*
* *XG Boost*
* *Decision Tree*

Below are the model accuracies for the 4 models mentioned above using the basic validation set approach,

|  |  |
| --- | --- |
| Model Fit | Model Accuracy |
| Logistic Regression | 80% |
| Random Forest | 83% |
| XG Boost | 85% |
| Decision Tree | 81% |

Based on the model accuracies, XGBoost is the considered the best model when compared to other models. Later, we scored the model on test data and got accuracy of 79% using XG boost model we fitted earier.

### Confusion Matrix for XGBoost Model on validation set approach



## Metrics

We created a function to calculate the model accuracy. The function can evaluate multiple metrics for evaluation namely

* *Log Loss – Sum of the logarithmic loss function*
* *AUC – Area under the Curve*
* *MAE – Mean Absolute Error*
* *MSE – Mean Square Error*
* *RMSPE – Root Mean Square Percentage Error*
* *Precision – True Positive / (True Positive + False Positive)*

\* *MAE, MSE and RMSE are not used for classification problems, they are used in continuous variable prediction.*

###################### Below code used in function for above metrics ####################

|  |
| --- |
| score **<-** **function(**a,b,metric**)**  **{**  switch **(**metric,  accuracy **=** sum**(**abs**(**a**-**b**)<=**0.5**)/**length**(**a**)**,  auc **=** auc**(**a,b**)**,  logloss **=-(**sum**(**log**(**1**-**b**[**a**==**0**]))** **+** sum**(**log**(**b**[**a**==**1**])))/**length**(**a**)**,  mae **=** sum**(**abs**(**a**-**b**))/**length**(**a**)**,  precision **=** length**(**a**[**a**==**b**])/**length**(**a**)**,  rmse **=** sqrt**(**sum**((**a**-**b**)^**2**)/**length**(**a**))**,  rmspe **=** sqrt**(**sum**(((**a**-**b**)/**a**)^**2**)/**length**(**a**))**  **)**  **}** |

The solution segment is divided into three parts :

* *The validation set approach*
* *Cross-Validation approach*
* *Model Ensembling*

## The validation set approach

The *Validation set approach,* involves dividing the available data randomly into training set, validation set and hold-out sample. The model fitting is done on the training set and this model is used to predict the response variable in the validation set. The resulting test error rate is assessed using the above metrics, either accuracy or LogLoss or precision etc. The main drawback of this method is:

* *The test error rate which is measured can be highly variable and depends on the observations used in the training data set and the ones used in the validation data set.*
* *In validation approach, basically the model is fit on a subset of observations called the training set and as we know the statistical methods tend to perform worse on fewer observations, there is a chance of over estimating the test error rate when the model is fit on the entire data set.*

## Cross-Validation approach

In the *Cross-Validation approach,* we perform the 5-fold cross validation on the training and validation data set.



Figure 5 Iteration of 5 fold cross validation[4]

*Below are the steps used to build cross validation for the data set*

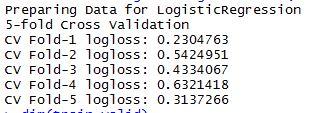
* *For each iteration, we take one subset out of the entire data set for validation and the remaining subsets are used for the training data.*
* *We build the model on training subset and scored the metrics on validation*
* *Last step is to pass the outcome of the results and probabilities of validation data to the score function to evaluate the model based on the metric we choose. In our case, we took the log loss metric for model evaluation.*

################# Below code is used for cross validation for logistic model #################

|  |
| --- |
| cat**(**cv, "-fold Cross Validation\n", sep **=** ""**)**  **for** **(**i **in** 1**:**cv**)**  **{**  Train\_Fold **<-** subset**(**Training, randomCV **!=** i, select **=** **-**c**(**order, randomCV**))**  Validation **<-** subset**(**Training, randomCV **==** i**)**    # building LogisticRegression model  model\_lr **<-** glm**(**result **~**., data**=**Train\_Fold, family**=**binomial**())**    # predicting on validation data  pred\_lr **<-** predict**(**model\_lr, Validation, type**=**"response"**)**  Validation **<-** cbind**(**Validation, pred\_lr**)**    # Printing CV scores  cat**(**"CV Fold-", i, " ", metric, ": ", score**(**Validation**$**result,  Validation**$**pred\_lr, metric**)**, "\n", sep **=** ""**)**  **}** |

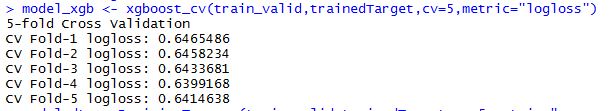
Similarly, we have used the cross-validation approach for the remaining models such as Random Forest, XG Boost, Decision Tree

### Results of 5-fold cross validation for logistic regression



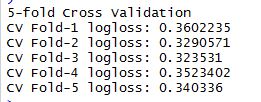
The logloss values are not consistent for logistic regression. Hence ,we do not choose this model

### Results of 5-fold cross validation for XGBoost



The logloss values are consistent for XGBoost which is better than logistic regression

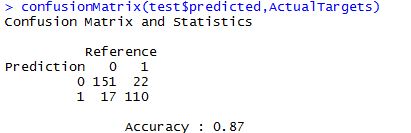
### Results of 5 fold cross validation for random forest



The logloss values are consistent as well as less when compared to XGBoost. Hence, we choose this model.

We scored the test data results on the random forest as shown below

### Confusion matrix for random forest on cross validation approach



We have chosen the random forest model based on the consistent cross validation scores on

the metric log-loss, and scored accuracy of 87% on the test data.

## Model Ensembling

Model Ensembling is a technique which is used to improve the accuracy of the models by combining two or more models .We are using a technique called simple majority vote ensemble to reduce the error rate and improve the model accuracy more efficiently.

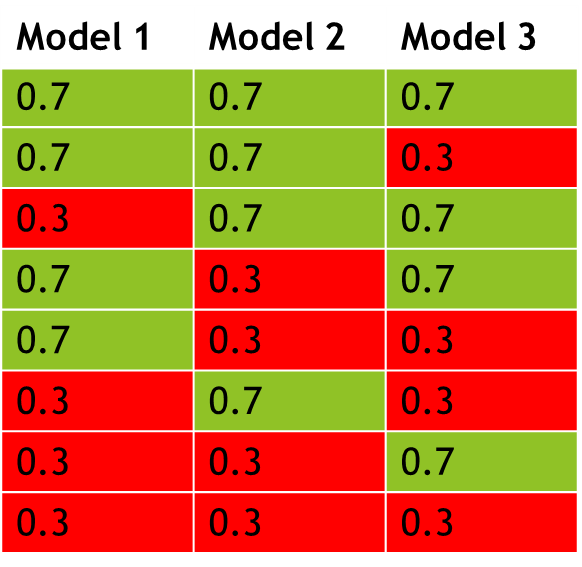
For example:

Consider a set of 10 samples with the actual values to be “1”

1111111111

Let us assume that we have three classifiers model1, model2, model3 with the accuracy of 70%,i.e 1 occurs 70% of the time and 0 occurs 30% of the times . So, we will have 8 different outcomes for the three binary classifiers with the majority vote as shown in the figure below.

|  |
| --- |
| **0.343** |
| **0.147** |
| **0.147** |
| **0.147** |
| **0.063** |
| **0.063** |
| **0.063** |
| **0.063** |



**Voting**

**Voting**

**0.784**

**0.784**

**0.216**

All three correct: 0.7 \*0.7\*0. = 0.343

Two correct: 0.7\*0.7\*0.3 + 0.7\*0.3\*0.7 + 0.3\*0.7\*0.7 = 0.441

Two wrong: 0.3\*0.3\*0.7 + 0.3\*0.7\*0.7 + 0.7\*0.3\*0.3 = 0.189

All three wrong: 0.3\*0.3\*0.3 = 0.027

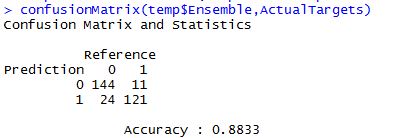
So, the average accuracy of majority vote ensemble is 78.4% ~ (0.343 + 0.441)

We used three models decision tree, random forest and xgboost for creating the ensemble model using majority voting i.e. if the predicted value is 1 in at least two models then the result is 1, otherwise it is 0.

################# Below code for Ensemble model using majority voting####################

|  |
| --- |
| cbind**(**ActualTargets,test**$**predict\_xgb,lr\_cv,dt\_cv,rf\_cv,xgb\_cv**)**  temp**=**as.data.frame**(**temp**)**  View**(**temp**)**  colnames**(**temp**)** **<-** c**(**"outcome","predict\_xgb",  "lr\_cv","dt\_cv","rf\_cv","xgb\_cv"**)**  temp**$**rf\_cv **=** as.numeric**(**rf\_cv**)**  temp**$**rf\_cv **=** temp**$**rf\_cv**-**1  temp**$**Ensemble **=** ifelse**(**temp**$**dt\_cv**+**temp**$**rf\_cv**+**temp**$**xgb\_cv **>=** 2,1,0**)**  confusionMatrix**(**temp**$**Ensemble,ActualTargets**)** |

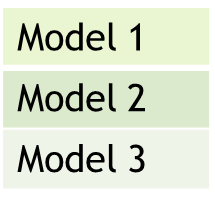
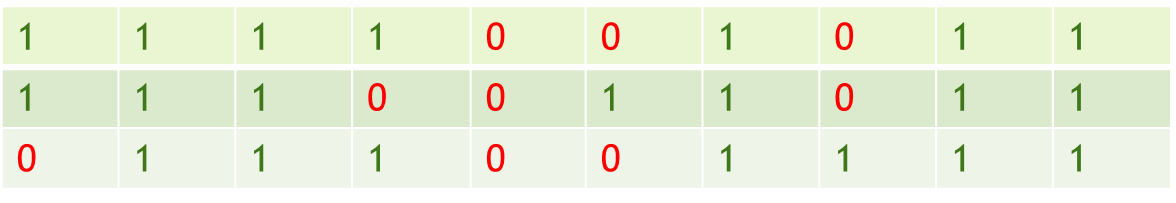
### Confusion Matrix for ensemble model using majority voting



The majority voting ensemble model is 88.33% accurate

# Limitations of Ensemble models using Voting

If the outcomes of the models being used for Ensemble modelling are correlated then we cannot improve the accuracy using Voting.



**0.70**

**0.70**

**0.70**

**Ensemble**

# 

**0.70**

If the outcomes of the are highly correlated, then ensemble modelling using voting fail to improve the accuracy.

# Other applications of cross validation

* Finding the value of lambda in regularizations
* For appropriate blending of ensemble models
* Feature selection

# References

[1] Validation methods. Outline Motivation Holdout set Re-sampling techniques Three-way data splits by Gilbert Butler

# [2] Cross-Validation for Genomic Prediction in SVS By [Bryce Christensen](http://blog.goldenhelix.com/author/bchristensen/)

<http://blog.goldenhelix.com/bchristensen/cross-validation-for-genomic-prediction-in-svs/>

# [3] Business Systems Intelligence: 5. Classification 2 Dr. Brian Mac Namee

[4] Institute for Genomics and Bioinformatics - Graz University of Technology  
Department of Information Design - FH JOANNEUM - Graz University of Applied Sciences

<https://genome.tugraz.at/proclassify/help/pages/XV.html>

# [5] KAGGLE ENSEMBLING GUIDE

<http://mlwave.com/kaggle-ensembling-guide/>