1. **Data description and sampling procedure**

The total number contains 1334 observations and 9 variables. Table 1 gives a list of variables in the data while identifying the variable type and their description**.** These variables are considered as potential predictors and are hence included in the model development.

Table 1

|  |  |
| --- | --- |
| altitude |  |
| preciptn |  |
| slope |  |
| roughness1 |  |
| aspect1 |  |
| tempAvg |  |
| tempMin |  |
| land |  |
| pb | **1 stands for presence and 0 for absence** |

For our analysis, we have randomly divided the data sample into two subsamples About two thirds of the whole data set (1001 observations) builds the first subsample, the TRAIN sample. It will be used for model estimation. The second subsample, TEST, with 333 observations will be used to get some overall procedure to compare methods and to check the predictive power of the models used.

**4．Model Selection and Development**

The most important thing in developing model is to select right modeling algorithm(s).

**4.1 Logistics regression**

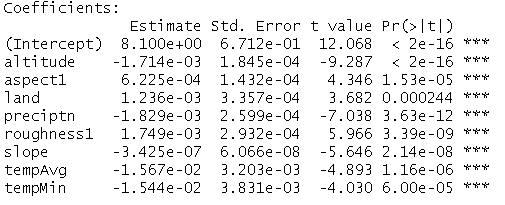
To select the best classifier, several options are used. The four options for the classifiers which were used are stated below.

1. Building the Logistic Model and checking the model summary

glm(formula = pb ~ ., data = training)

1. Significance Level of the Variables

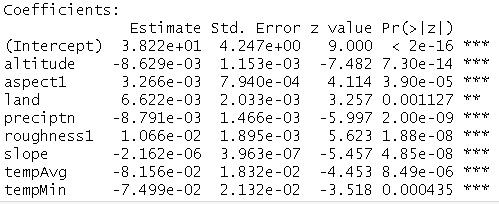
Those variables which have at least one star in the coefficients table are significant. Positive coefficient means higher the value of that variable, an increased risk of default, and vice versa. All variables are significant.



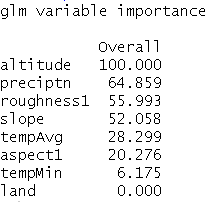
1. Using the dataset in the caret package.

Using the training hornbill dataset, which contains *1001* observations, we will use logistic regression to model *pb* as a function of eight predictors.

mod\_fit=train(pb~.,data=training,trControl=train\_control,method="glm",family="binomial")



1. Using caret check importance of the different predictors

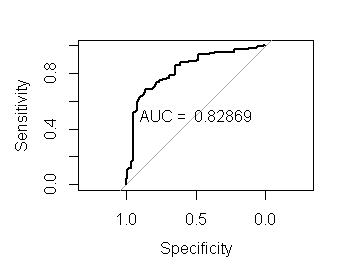


1. test the model

predict on the test data

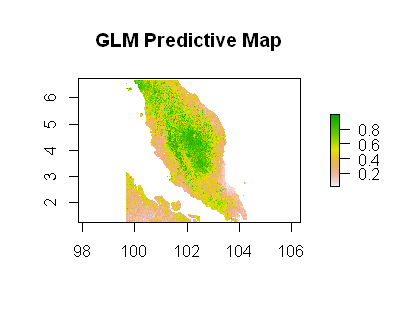
1. test model fit-auc

The receiving operating characteristic is a measure of classifier performance. Using the proportion of positive data points that are correctly considered as positive and the proportion of negative data points that are mistakenly considered as positive, we generate a graphic that shows the trade off between the rate at which you can correctly predict something with the rate of incorrectly predicting something. Ultimately, we’re concerned about the area under the ROC curve, or AUROC. That metric ranges from 0.50 to 1.00, and values above 0.80 indicate that the model does a good job in discriminating between the two categories which comprise our target variable. Bear in mind that ROC curves can examine both target-x-predictor pairings and target-x-model performance. An example of both are presented below.



1. build an SDM

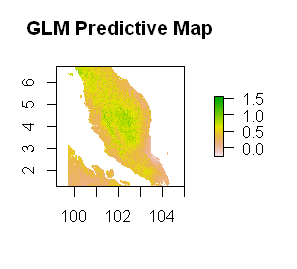
read in all predictors-since they are all significant, use predict to implement the GLM model stored in mod\_fit on the raster stack of our predictors



this is predictive map and the darker colors the green colors means higher chances of finding a given species or better habitat suitability for the different hornbill species we are considering.

1. use the basic GLM to predict

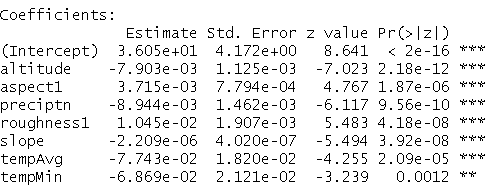
implement the GLM model stored in mod\_fit on the raster stack of our predictors



1. remove land use as a predictor

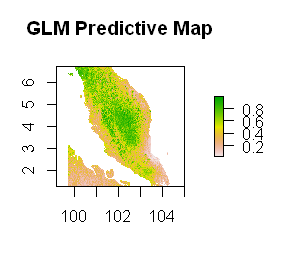
mod\_fit2=train(pb~altitude+aspect1+preciptn+roughness1+slope+tempAvg+tempMin

,data=training,trControl=train\_control,method="glm",family="binomial")



1. use predict to implement the GLM model stored

in mod\_fit on the raster stack of our predictors

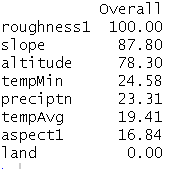


**4.2 support vector machine**

**1) svm with rbf kernel**

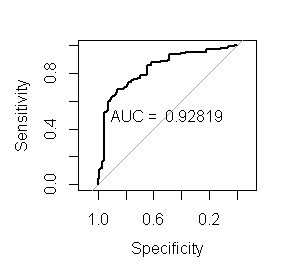


**2) importance of the different predictors**



**3) test the model**

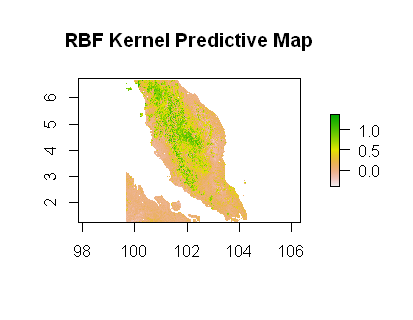
**4) Test model fit-auc**



**5) read in all predictors**

**use predict to implement the SVM model stored**

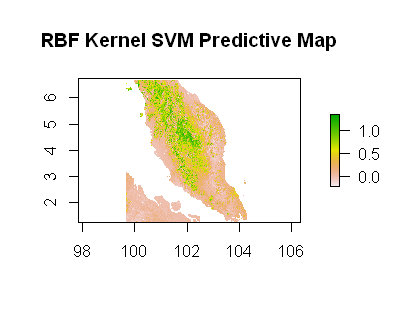
**#in mod\_fit on the raster stack of our predictors**



**6) drop land**

**mod\_fit2=train(pb~altitude+aspect1+preciptn+roughness1+slope+tempAvg+tempMin,**

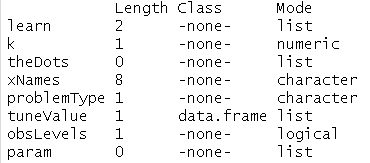
**data=pa,trControl=train\_control,method="svmRadial")**



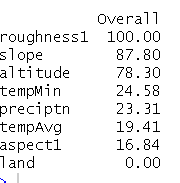
seem to be a bit darker indicating more suitability here and a bit more stability

4.3 KNN

**1) KNN mod\_fit=train(pb~.,data=training,trControl=train\_control,method="knn")**

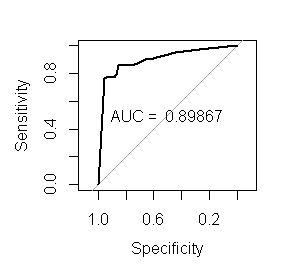


**2) importance of the different predictors**



**3) test the model**

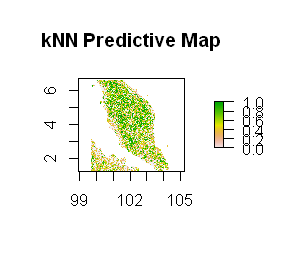
**4) Test model fit-auc**



**5) read in all predictors**

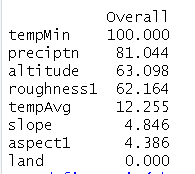
**use predict to implement the KNN model stored**

**#in mod\_fit on the raster stack of our predictors**



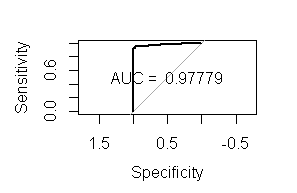
4.4 random forest

**2) importance of the different predictors**



**3) test the model**

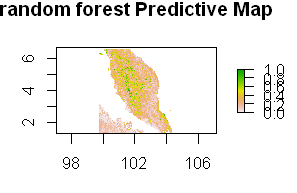
**4) Test model fit-auc**

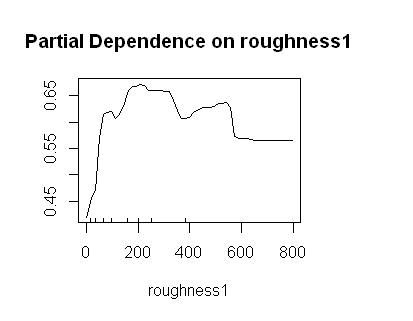


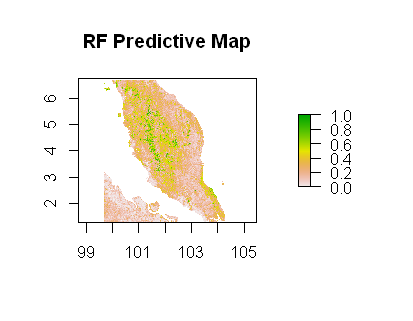
**5) read in all predictors**

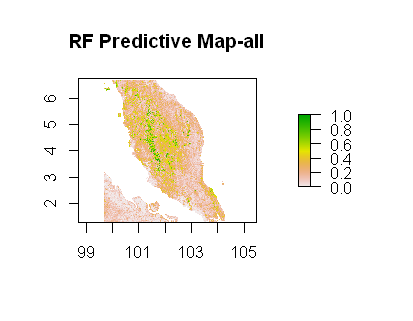
#use predict to implement the MARS model stored

#in mod\_fit on the raster stack of our predictors









 Gradient Boosting Machine (GBM)

