Cheat Sheet – 10 Machine Learning Algorithms & R Commands

This article lists down 10 popular **machine learning algorithms** and related **R commands** (& package information) that could be used to create respective models. The objective is to represent a quick reference page for beginners/intermediate level R programmers who working on machine learning related problems. Please feel free to comment/suggest if I missed to mention one or more important points. Also, sorry for the typos.
Following are the different ML algorithms included in this article:

1. Linear regression
2. Logistic Regression
3. K-Means Clustering
4. K-Nearest Neighbors (KNN) Classification
5. Naive Bayes Classification
6. Decison Trees
7. Support Vector Machine (SVM)
8. Artifical Neural Network (ANN)
9. Apriori
10. AdaBoost

Cheat Sheet – ML Algorithms & R Commands

- **Linear regression**: "lm" method from base package could be used for linear regression models. Following is the sample command:

```
lm_model <- lm(v ~ x1 + x2, data=as.data.frame(cbind(y,x1,x2)))
```

- **Logistic Regression**: Logistic regression is a classification based model. "glm" method from base R package could be used for logistic regression. Following is the sample command:

```
glm_model <- glm(v ~ x1+x2, family=binomial(link="logit"), data=as.data.frame(cbind(y,x1,x2)))
```

- **K-Means Clustering**: “kmeans” method from base R package could be used to run k-means clustering. Following is a sample command given X is a data matrix and m is the number of clusters:

```
kmeans_model <- kmeans(x=X, centers=m)
```

- **K-Nearest Neighbors (KNN) Classification**: “knn” method from “class” package could be used for K-NN modeling. One need to install and load “class” package. Following is the sample command given X_train represents a training dataset, X_test represents test data set, k represents number of nearest neighbors to be included for the modeling

```
knn_model <- knn(train=X_train, test=X_test, cl=as.factor(labels), k=K)
```

- **Naive Bayes Classification**: “naiveBayes” method from “e1071” package could be used for Naive Bayes classification. One need to install and load “e1071” package prior to analysis. Following is the sample command:

```
naiveBayes_model <- naiveBayes(y ~ x1 + x2, data=as.data.frame(cbind(y,x1,x2)))
```

- **Decision Trees**: “rpart” method from “rpart” can be used for Decision Trees. One need to install and load “rpart” package. Following is the sample command:

```
cart_model <- rpart(y ~ x1 + x2, data=as.data.frame(cbind(y,x1,x2)), method="class")
```

- **Support Vector Machine (SVM)**: “svm” method from “e1071” package could be used for SVM. Note that the same package also provide method, naiveBayes, for Naive Bayes classification. One need to install and load “e1071” package. Following is the sample command given X is the matrix of features, labels be the vector of 0-1 class labels, and C being regularization parameter

```
svm_model <- svm(x=X, y=as.factor(labels), kernel ="radial", cost=C)
```

- **Artifical Neural Network (ANN)**: “neuralnet” method from “neuralnet” package could be used for ANN modeling. Following is sample command:

```
ann_model <- neuralnet( y ~ x1 + x2 + x3, data=as.data.frame(cbind(y,x1,x2,x3)), hidden = 1)
```

  Prediction could be made using following formula:

```
p <- compute( ann_model, as.data.frame(cbind(x1,x2))
)
```

- **Apriori**: “apriori” method from “arules” package could be used for Apriori analysis. One need to install and load “arules” package. Following is the sample command:

```
apriori_model <- apriori(as.matrix(sampleDataset), parameter = list(supp = 0.8, conf = 0.9))
```

- **AdaBoost**: “ada” method from “rpart” package could be used as boosting function. Following is sample command:

```
boost_model <- ada(x=X, y=labels)
```

For most of the above formulas including linear regression model, one could use following function to predict:

```
predicted_values <- predict(some_model, newdata=as.data.frame(cbind(x1_test, x2_test)))
```