Brief Cheat Sheet on Machine Learning

Thiloshon Nagarajah

April 29, 2017

Drum Roll

This is a brief cheat sheet for machine learning process pipeline. This is by no means elaborate, just listing the concepts i learned from the Coursera course structurely to get familiar with. The content is from and only from the Practical Machine Learning course of Data Science specialization of Coursera, so the credit goes to the awesome instructors and relevant sources.

Introduction

Most of the Machine Learning algorithms and various other concepts were created by various people. So to use these, you need to use relevent packages. To make the job easier, a unifying framework has been created called `caret`. This package has almost everything you need in machine learning.

The important aspects of machine learning in order of importance is,

Question > Data > Features > Algorithms

Everything on the right hand side depends heavily on things on the left hand side. Giving much thought to earlier parts makes the later part easy and effective.

Any data has signal and noise. Machine learning and defining predictors is to capture the signal.

Practical Implementation in Recommended Chronological Order

1. Splitting

The first thing to do is, splitting the data to number of subsets.

1. Training (60% of data)
2. Test (20% of data)
3. Validation (Optional, 20% of data)

On splitting data there is a concept called Cross Validation. It is converting training set to a test/train set and averaging results before applying to test. It helps in

1. estimating out-of-sample accuracy rate
2. picking variables to include in model

3. picking type of prediction function
4. picking parameters in the prediction function
5. and to compare different predictors

In caret `createdatapartition` is used to split data.

Common types of Cross validating:

1. Random subsampling - It is mostly done without replacement, otherwise it's called bootstrap: Problem with bootstrap is it tends to underestimate error. But there are workarounds which are widely used. In caret, `createresample` is used.
2. k fold - If you choose small k, you are likely to introduce bias while large k introduces variance. In caret, `createFolds` is used.
3. leave one out

`createtimeslices` is also used for specific needs.

2. Exploratory Analysis

Exploratory analysis is very important step in understanding the data and understanding features. It shows major trends or patterns in data without much hassle, shows imbalance in outcomes/ predictors, outliers, skewed variables and groups of points that are not explained by a predictor.

1. Use `summary`, `str` ucture and so on to understand basic dataset properties.
2. Use plots to observe interesting patters or linearity or non-linearity.
3. Featureplot `featurePlot` to get understanding on top level patterns.
4. Then `qlopt` or `ggplot` interesting variates from Feature plot. Make use of color, regression smoothers `geom_smooth`, etc to better analyse.
5. Make factor variables from continuous variables to understand more. `cut2` in `Hmisc` package is useful.
6. Use `table`, `jitter`, density plots, etc too.

3. Pre-processing

Most of the time pre procesing is done to make predictions accurate. Mainly, pre processing is a mechanism of cleaning data and also compressing data without loss of features.

1. Standardizing When deviance is high in certain variates, standardizing helps reduce deviance. `caret` has methods and parameters to tweak standardizing methods. `center`, `scale` are two main method parameters used. Box-Cox transform is a standardizing method to normalize continuous data using maximum likelihood. imputing data is also used. k-nearest-neighbors is the mainly used technique in imputing. `knnImpute` is used in `caret`.

2. Covariate Creation At times it helps to make new co variates to get a better prediction. Raw data is used to create covariate and at times transforming tidy covariate to new variates is also done. Factor variables are converted to new indicator variables using `dummyVars` Removing zero covariates is done with `nearZeroVar` Spline basis is another method of covariate creation. `bs` from `splines` package or `gam` in `caret` can be used.

3. Pre-processing with Principal Component Analysis Weighted combination of predictors are used to make a better prediction model. This greatly helps in data compression too. Correlated predictors are found with `cor` Principal Componant Analysis (PCA) is the Right singular value from Singular Value Decomposition (SVD). `prcomp` in `caret` or else `method="pca"` in `preProcess` can be used.

All these pre processings can be configured in the `train` method of `caret` . In addition `train control` parameter can be set too. Bootstrap(`boot`) or cross validation(`cv`), `repeatedcv` , `LOOCV` methods can also be set.

Further Metric options can be set to the train function call. It takes following parameters

1. For Continuous outcomes: RMSE, RSquared
2. For categorical outcomes: accuracy, kappa

4. Prediction Models and Predictions

Prediction models or algorithms should be

1. accurate
2. scalable
3. interpretable
4. simple
5. fast

Varoius prediction models

1. Predicting with linear regression. method is set as `lm`
2. Predicting with trees. Evaluate homogeneity within groups. method is set as `rpart` The measures of impurity in this model are,

a. Misclassification Error
b. Gini index
c. Deviance/information gain. To visually plot the tree, `plot` or `fancyRpartPlot` from package `rattle` can be used.

3. Bootstrap Aggregating (bagging). Averaging complex models to get balanced and accurate model. bagged loess is created which is similar to spline curves. In caret, method= `bagEarth` , `treebag` , `bagFDA` or your custom method can be used.
4. Random Forests. method is set as `rf`
5. Boosting. Weighting weak predictors. Similar to bagging or random forests. Number of ways can be used in R, `gbm` : boosting with trees `mboost` : model based boosting `ada` : statistical boosting based on additive logistic regression `gamBoost` : boosting generalized additive models
6. Model Based Prediction.

7. Regularized Regression. fiting linear regression or generelized linear regression and then penalizing or shrinking large coefficients. lasso regularized regression can help with model selection.
8. combining predictors. Bagging, Boosting and RF are combining similar classifiers. Model Stacking, Model Ensembling are combining different classifiers
9. Forecasting and Time Series Prediction can be complicated since data are dependent over time. Test/Train splitting, correlation and extrapolation has to be done carefully.
10. Unsupervised Prediction. Creating, naming clusters and building predictors from the clusters. k-means clustering is mainly used. `cl_predict` in package `clue` can be used.

5. Model Comparisons

`confusionMatrix` is widely used to compare models or quantify accuracy, sensitivity and specificity.

There are number of error definition that can be used.

Common error measures:

1. Mean Squared Error(MSE)
2. Median absolute deviation
3. Sensitivity
4. Specificity
5. Accuracy
6. Concordance (kappa is the common one)
7. For continuous data - Mean Squared Error and Root Mean Squared Error

and also,

8. false positive, false negative
9. positive predictive value
10. negative predictive value

In Sample Error (Resubstitution error). The error rate you get in the same data set you used to build predictor. Out of Sample Error (Generalization error). The error rate you get in new data sets.

ROC (Receiver Operating Characteristic curves). How good the prediction algorithm is. Area Under Curve (AUC) of FalsePositive ~ TruePositive curve 1 is perfect classifier, 0.8 above is good classifier.