README for the classification tree

Group 2

December, 2nd, 2019

Abstract

This Chapter is describes the classification tree algorithm applied to the different data sets (all with '_3', one unsampled and four sampled), coded in the script 'ClassificationTree.R'.

Fundamentals of Classification Trees

Classification trees split the different variables in order to obtain the most homogeneos possible clusters, by minimizing a loss function, that can be restricted (this complexity parameter is called 'cp' in the rpart-Package). For every split it computes the sum of the errors on both sides of the split for all Variables and chooses the one with the lowest error.

Our Approach

As also applied in the other models, we use all the available information just before the 2014 NFL-Draft, in order to train the model and then apply it on the data for 2014. In other words we act as if it was the end of April 2014 (which is one week before the draft).

For growing trees on our College League / NFL Draft data, we check whether the best results can be optained, by manually splitting the data sets on the three postitions (QB / WR / RB) or if the computer will do that on his own. For growing the trees we use the rpart-Package, which is commonly used for this purpose, since it does very much on his own. When growing a tree it uses k-fold cross-validation (by default k=10) for optimizing the model with respect to the best complexity and the spots to split. Therefore we do no further cross-validation on the data set.

You can see the different trees, that we grew, plotted with the fancyRpartPlot-function out of the rattle-Package in the Appendix at the end of this file. We are cross-validating the sampling methods and therefore grow 20 trees in total, we will still just display the four trees from the unsampled dataframe. Since we use data with many variables and a couple of splits are made, the plots are not really readable. The aim of showing them, is to visualize the complexity of the trees.

Performance Measurement of the trees

We now want to have a look on the performance of the trees. In order to compare the results to other models performance, we decided to always take unsampled data for this. And since our business case is to predict the 2014 draft (as if it was the day before), we apply it on the data from 2007 to 2013 to compare the classification errors.

Table 1: True Positives/Negatives and False Positives/Negatives of the trees for the different models on training data

	No Sampling	Oversampling	Undersampling	Rose Both	Smote
QB_TP	46	68	70	65	65
QB_TN	327	292	221	276	281
QB_FP	10	45	116	61	56
QB_FN	27	5	3	8	8
WR_TP	79	146	152	149	138
WR_TN	1118	948	921	889	910
WR_FP	18	188	215	247	226
WR_FN	85	18	12	15	26

	No Sampling	Oversampling	Undersampling	Rose Both	Smote
RB_TP	48	80	74	81	81
RB_TN	527	462	437	425	412
RB_FP	11	76	101	113	126
RB_FN	42	10	16	9	9
Together_TP	116	291	290	289	271
Together_TN	1974	1563	1566	1522	1599
Together_FP	37	448	445	489	412
Together_FN	211	36	37	38	56

Table 2: True Positives/Negatives and False Positives/Negatives of the trees for the different models on testing data

	No Sampling	Oversampling	Undersampling	Rose Both	Smote
QB_TP	8	9	10	9	9
QB_TN	93	79	63	73	78
QB_FP	12	26	42	32	27
QB_FN	3	2	1	2	2
WR_TP	4	21	21	21	21
WR_TN	315	263	257	251	261
WR_FP	21	73	79	85	75
WR_FN	18	1	1	1	1
RB_TP	6	10	11	10	12
RB_TN	172	153	142	131	121
RB_FP	10	29	40	51	61
RB_FN	8	4	3	4	2
Together_TP	10	41	42	37	43
Together_TN	593	461	457	444	460
Together_FP	30	162	166	179	163
Together_FN	37	6	5	10	4

In order to compare them better, we will now have a look at the ratio of correct classification, which is equal to:

$$\frac{CorrectClassifications}{AllClassifications} = \frac{TP + TN}{TP + TP + FP + FN}$$

Table 3: Percentage of right classifications on training data

Sampling	QB	WR	RB	Together
no_sampling	0.9098	0.9208	0.9156	0.8939
oversampling	0.8780	0.8415	0.8631	0.7930
undersampling	0.7098	0.8254	0.8137	0.7938
Rose_both	0.8317	0.7985	0.8057	0.7746
Smote	0.8439	0.8062	0.7850	0.7998

Table 4: Percentage of right classifications on testing data

Sampling	QB	WR	RB	Together
no_sampling	0.8707	0.8911	0.9082	0.9000
oversampling	0.7586	0.7933	0.8316	0.7493
undersampling	0.6293	0.7765	0.7806	0.7448
Rose_both	0.7069	0.7598	0.7194	0.7179
Smote	0.7500	0.7877	0.6786	0.7507

Conclusion

As we see, the models for the manually separated positions mostly perform better than the model for QB/WR/RB together. But the bigger effect, which we can see, is that the sampling reduces the accuracy of the models quite much. Since we have 12.4% of drafted players in the data, a model predicting 0 (=not drafted) would outperform the models trained on sampled data.

Within the Classification tree models, the highest accuracy could be obtained, by using the three models for the manually separated positions with a weighted average accuracy of 91.74%. Keeping in mind, that a model that always predicts '0' would also have an accuracy of 87.6%, this models performance, which looks very good on the first sight, is only a quite small improvement.

Appendix

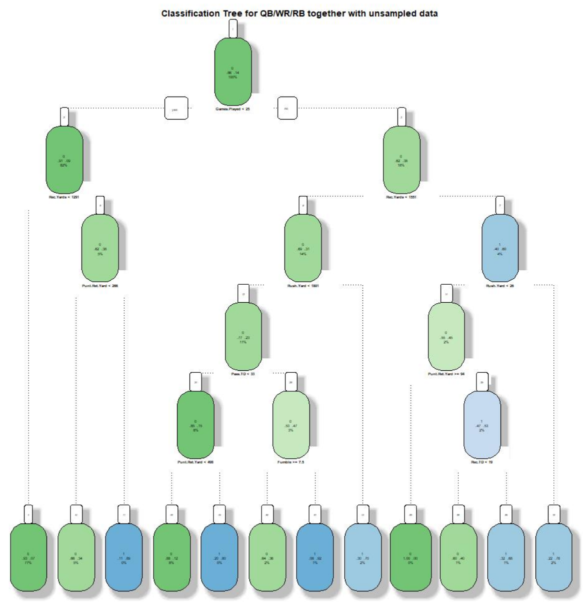

Figure 1: Tree for QB/WR/RB together on unsampled data

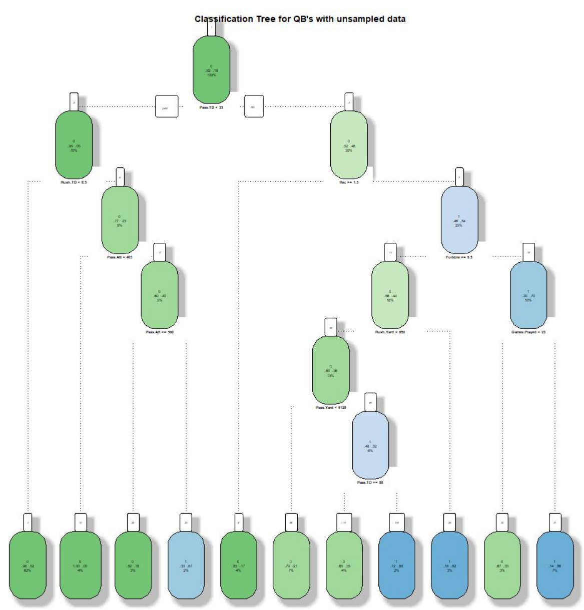

Figure 2: Tree for QB's on unsampled data

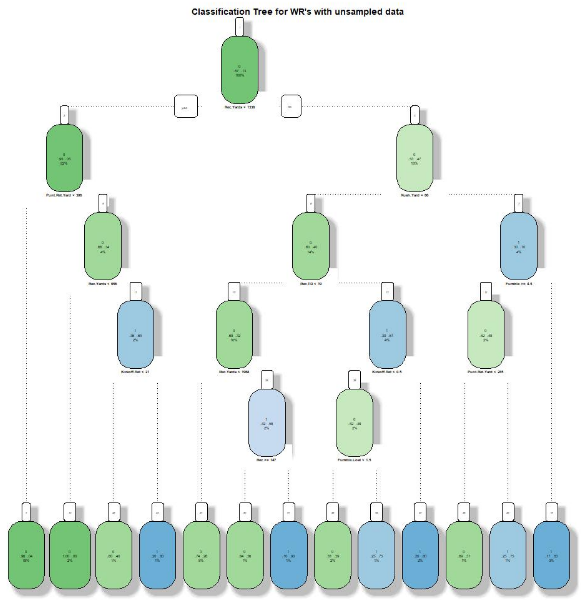

Figure 3: Tree for WR's on unsampled data

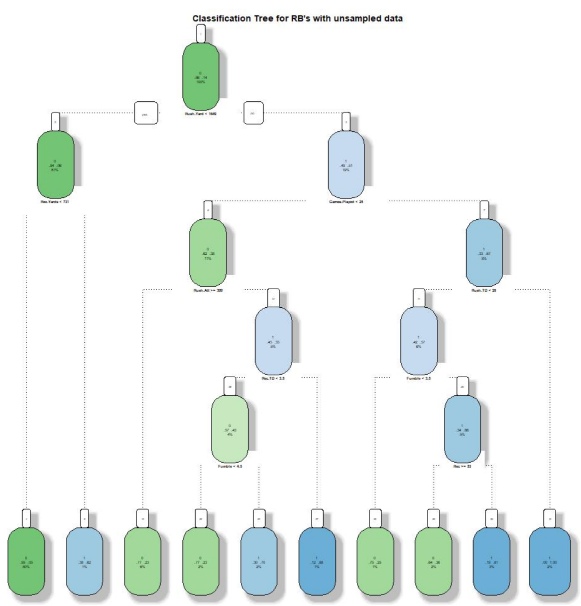

Figure 4: Tree for RB's on unsampled data