Tutorial Business Analytics

Exercise 8

Exercise 8.1

True Class	Predicted Class
0	0
0	1
1	1
1	0
0	0
1	0
0	0
1	1
0	1
1	0

Calculate Recall, False Alarm Rate, Precision, Specificity and Accuracy.

Solution 8.1

True Class	Predicted Class	
0	0	**TN**
0	1	**FP**
1	1	**TP**
1	0	**FN**
0	0	**TN**
1	0	**FN**
0	0	**TN**
1	1	**TP**
0	1	**FP**
1	0	**FN**

Recall (True Positive Rate, Sensitivity, Hit Rate)
"How many positive instances have been predicted to be positive"

$$tpr = \frac{tp}{tp + fn} = \frac{2}{2+3} = 0.4$$

False Alarm Rate (False Positive Rate)
"How many negative instances have been predicted to be positive"

$$fpr = \frac{fp}{fp + tn} = \frac{2}{2+3} = 0.4$$

Precision (Positive Predictive values)
"How many positively predicted instances have been positive"

$$pre = \frac{tp}{tp + fp} = \frac{2}{2+2} = 0.5$$

Specificity (True Negative Rate)
"How many negative instances have been predicted to be negative"

$$tnr = \frac{tn}{fp + tn} = \frac{3}{2+3} = 0.6 = 1 - fpr$$

Accuracy
"How many instances have been predicted correctly"

$$acc = \frac{tp + tn}{tp + fp + tn + fn} = \frac{2+3}{2+2+3+3} = 0.5$$

Exercise 8.2

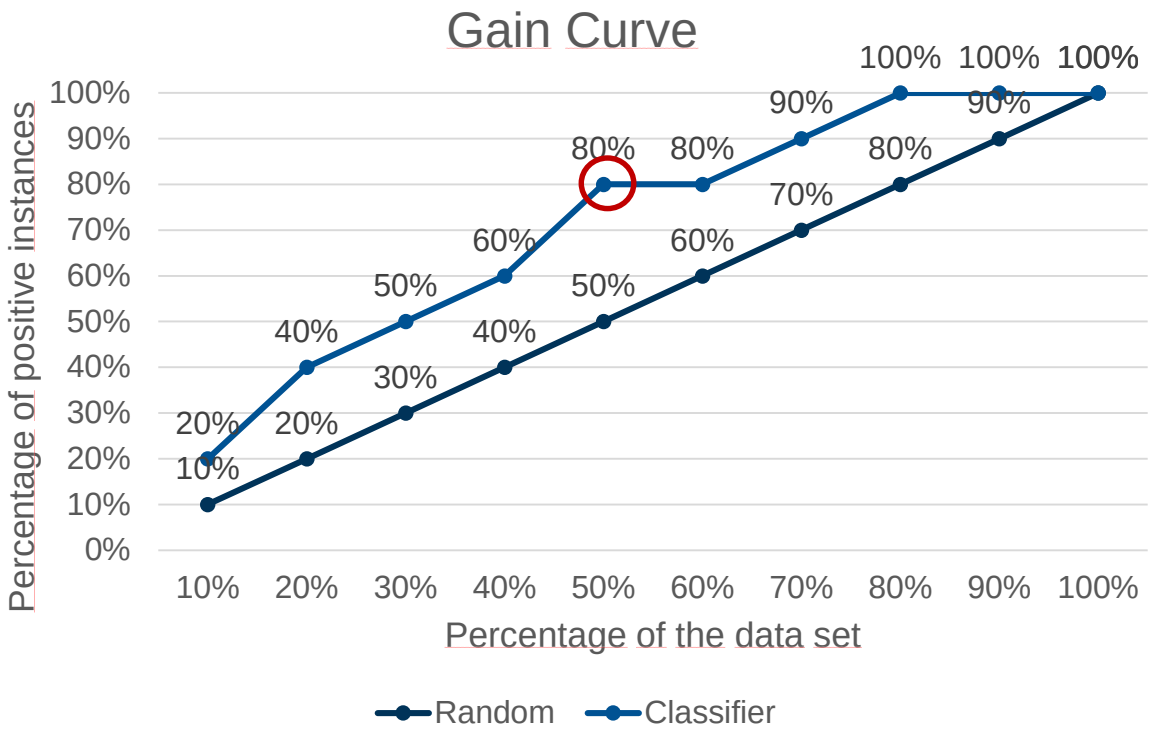
Use the given result of an evaluation (Cutoff = 0.87) to construct:

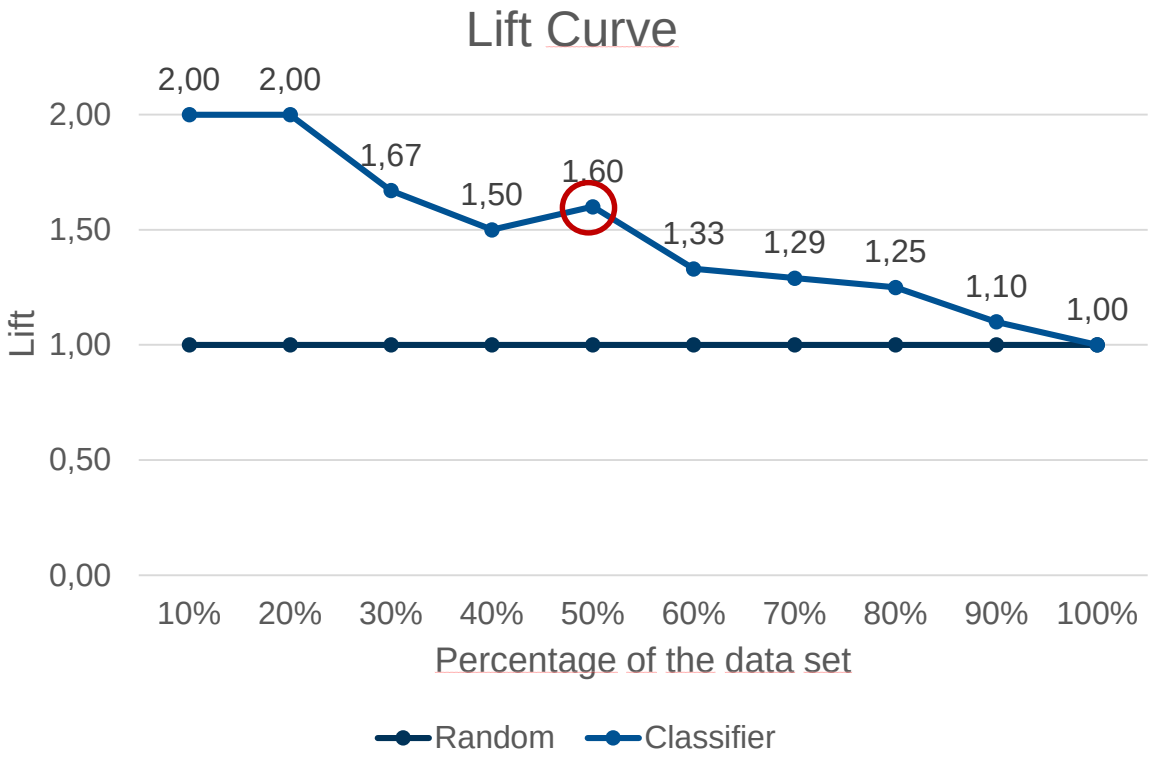
- a gain curve (10% steps)
- a lift curve
- an ROC curve

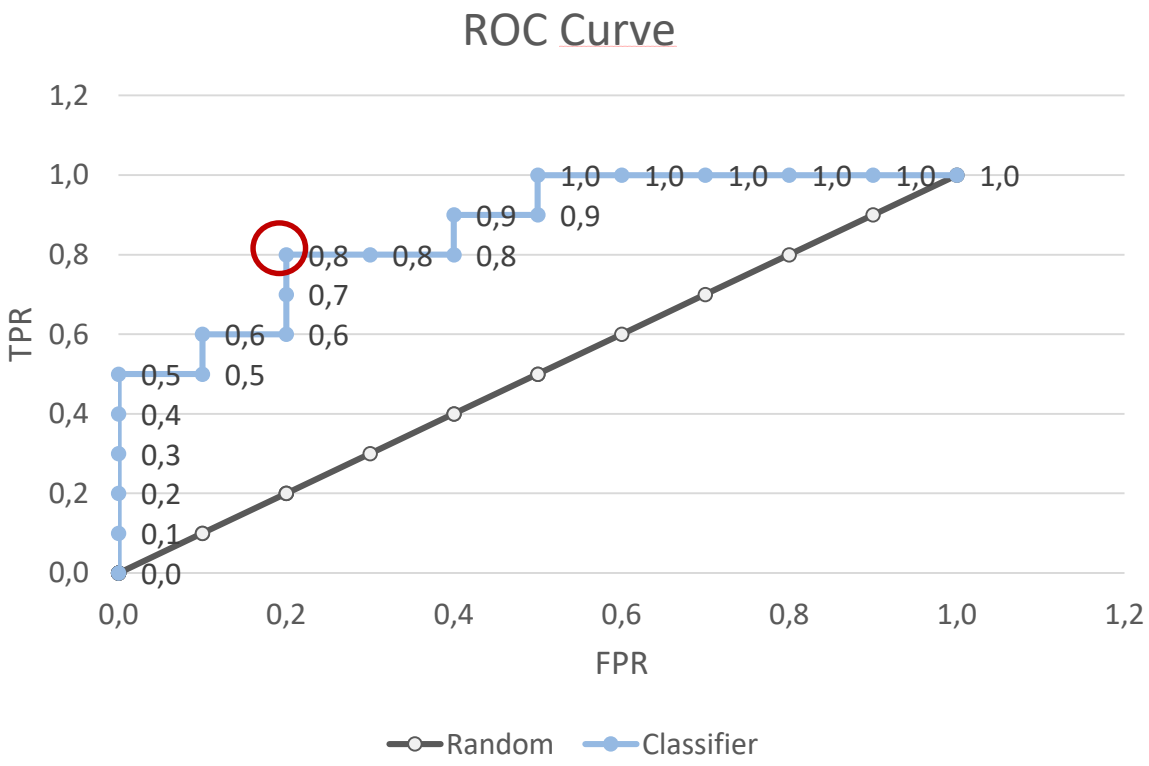
Remember: A Cutoff value of 0.87 means, we will classify an instance as positive until its probability falls under 0.87

Number	Probability	Class
1	0.991	+
2	0.977	+
3	0.973	+
4	0.945	+
5	0.918	+
6	0.915	-
7	0.906	+
8	0.889	-
9	0.873	+
10	0.871	+
11	0.869	-
12	0.866	-
13	0.862	+
14	0.852	-
15	0.837	+
16	0.831	-
17	0.829	-
18	0.811	-
19	0.787	-
20	0.779	-

Solution 8.2


Gain Curve
Percentage of positive instances
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
Percentage of the data set
Random
Classifier


Lift Curve
Lift
2,00
1,50
1,00
0,50
0,00
10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
Percentage of the data set
Random
Classifier


ROC Curve
TPR
FPR
1,2
1,0
0,8
0,6
0,4
0,2
0,0
0,0
0,2
0,4
0,6
0,8
1,0
1,2
Random
Classifier

Exercise 8.3

Working for a large e-commerce enterprise, you are given a dataset for customer churn prediction. The xlsx-File “E Commerce Dataset” contains a description of all variables in sheet “Data Dict” as well as the raw data itself in sheet “E_Comm”.

You intend to find a good random forest model to predict future customer churns.

Note: Use the R-Script “8_3_Churn_Prediction.R”

You begin with some data preparations. You remove all cases with missing data and factorize all variables where necessary.

a. Your colleague proposes to train the model on the entire dataset and argues to tune the “mtry” and “trees” parameters until the training accuracy is maximized. Do you agree? If not, which issues can you identify with this approach?

Your colleague has little understanding of the issues that you raised. You decide to illustrate your ideas by means of the first twenty instances in your data {1, 2, …, 20}.

b. You decide to split your dataset into training and test set with a 80%-20% split and perform a 4-fold cross-validation. Using the small dataset, design an exemplary split of the data. Show how you would partition the data for the 4-fold cross validation. Explain the purposes of each subset and which operations / actions you perform with each subset.

You now turn back to the original dataset.

c. Perform training, 4-fold cross-validation and testing with a 60%-20%-20% split in R. Use the precision as metric for model selection. Build a confusion matrix for the test set and report precision, accuracy, and recall.

d. On another dataset, you notice missing values for some numeric attributes in the test set. Your colleague suggests to impute these missing values by the mean of this attribute across all test instances. Do you agree? Explain your reasons.

Solution 8.3

a) See *8_3_Churn_Prediction.R*
This approach will result in an overfitted model with high variance. This means that the model will probably not be able to generalize to the population. Hence, when our model is given a new dataset which is different from the training data but still from the same population, it will likely perform poorly.

b) An exemplary solution will be:

- **Training/Test-Split:**
  Training Set: {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16}
  Test Set: {17,18,19,20}

- **4-Fold Cross-Validation:**

1. Fold: {1,2,3,4}
2. Fold: {5,6,7,8}
3. Fold: {9,10,11,12}
4. Fold: {13,14,15,16}

1. Iteration:
   Training Set: {1,2,3,4,5,6,7,8,9,10,11,12}
   Validation Set: {13,14,15,16}

2. Iteration:
   Training Set: {1,2,3,4,5,6,7,8,13,14,15,16}
   Validation Set: {9,10,11,12}

3. Iteration:
   Training Set: {1,2,3,4,9,10,11,12,13,14,15,16}
   Validation Set: {5,6,7,8}

4. Iteration:
   Training Set: {5,6,7,8,9,10,11,12,13,14,15,16}
   Validation Set: {1,2,3,4}

- During each iteration of the cross-validation, a learning model (in this case a decision tree) will be fitted on the training data of the corresponding iteration and will be validated on the validation set of the iteration. For each iteration, the value of the chosen evaluation metric (accuracy, recall, F1 score, etc.) will be calculated, and the average of those values will be to select an appropriate model.
- The test set can be thought as a proxy of unseen data, and it will not be used until the very end of the modeling phase. It will only be used after we are satisfied with our model selection. Applying the model on

the test set and calculating the evaluation metric allows for an assessment of the overall goodness of the model.

c) See *8_3_Churn_Prediction.R*

d) Imputing the missing values with the mean of the dataset is a common approach. However, the imputation should always be based on the mean of the training set, and not on the test set. The test set is a proxy for unseen data and therefore it may not be used for imputation.

This also means that the imputation should be done after the train-test-split, since imputing the missing data from the full dataset will expose information about the test data into the training data, causing the phenomenon referred to as data leakage. This will result in overfitting and having an overly optimistic evaluation of the model's performance on unseen data.