Online Test 2: Classification

Test und Assessment – Druckansicht

Online Test 2: Classification

Datum: Sun Mar 13 14:12:01 2022 Maximale Punktezahl: 13

Frage 1 - Classification Models - Structure (1 Punkt) [ID: 1063742]

Match each classification model to its structure!

No model	passt zu	k-NN	*(0.2 Punkte)*
Hyperplane	passt zu	Linear model	*(0.2 Punkte)*
Hyperplane, possibly in transformed space	passt zu	SVM	*(0.2 Punkte)*
Axis-parallel hyperplanes	passt zu	Decision tree	*(0.2 Punkte)*
Probability distribution	passt zu	Naive Bayes	*(0.2 Punkte)*

See the lecture.

Frage 2 - Classification Models - Weaknesses (1 Punkt) [ID: 1063749]

Match each classification model to a core weakness!

Needs to query training dataset for prediction.	passt zu	k-NN	*(0.2 Punkte)*
Coefficients might become meaningless if predictors are correlated.	passt zu	Linear model	*(0.2 Punkte)*
Parametrization (for appropriate shape of decision boundary) difficult.	passt zu	SVM	*(0.2 Punkte)*
Likely to overfit.	passt zu	Decision tree	*(0.2 Punkte)*
Needs to assume a distribution for numeric features.	passt zu	Naive Bayes	*(0.2 Punkte)*

See the lecture.

Frage 3 - Decision Trees - Training Complexity (1 Punkt) [ID: 1051787]

Given a dataset with m data objects and n features, what is the time complexity of training a decision tree, assuming it is balanced?

- ○ $O(m * n)$
  (0 Punkte)
- ○ $O(m * n^2)$
  (0 Punkte)
- ◉ $O(m * \log m * n)$
  (1 Punkt)

- ○ $O(m * \log m * n * \log n)$
  (0 Punkte)
- ○ $O(m^2 * n^2)$
  (0 Punkte)

A balanced tree for m data objects has a depth of $\log m$. For each node, all m data objects (further down in the tree there will be fewer) and n features need to be traversed to determine the next split. The feature values should be sorted, but this can be done as a one-time preliminary effort in $O(m * \log m * n)$.

Frage 4 - Decision Trees - Prediction Complexity (1 Punkt) [ID: 1051788]

Given a balanced decision tree trained on a dataset of m data objects and n features, what is the time complexity of the prediction for one new data object?

- ○ $O(1)$
  (0 Punkte)
- ◉ $O(\log m)$
  (1 Punkt)
- ○ $O(m)$
  (0 Punkte)
- ○ $O(m * n)$
  (0 Punkte)
- ○ $O(m * \log m * n)$
  (0 Punkte)

A balanced tree has a depth of $\log m$. This tree has to be traversed downwards for a prediction. The number of features is no longer relevant, because only one feature is checked in each split.

Frage 5 - Decision Trees - Prediction Quality (1 Punkt) [ID: 1051789]

You train a decision tree without any pre- or post-pruning (i.e., the tree can branch an arbitrary number of times) on an arbitrary dataset with two classes and an arbitrary split in training data and test data.

In the worst case, the accuracy can be [50%] *(0.5 Punkte)* on the training data and [0%] *(0.5 Punkte)* on the test data.

Lücke 1: Quite often, decision trees trained without any limitations can easily reach perfect or nearly perfect accuracy on the training data, as they can just separate objects until the partitions are pure regarding the classes. The accuracy on the training data can fall below 100%, though, even if the training is done without limitations. That is the case if data objects belonging to different classes cannot be discriminated using the given feature values. This is called the irreducible error or "Bayes Error". For binary classification and assuming balanced classes (frequency 50:50), this means that 50% of data objects are classified wrongly and therefore only 50% are classified correctly. If, however, the classes are present in different proportions, the accuracy can be above 50% for non-discriminatory features due to guessing the more frequent class.

Lücke 2: If the features on the training data are non-discriminatory, the decision tree can only guess on the test data either. Apart from that, it is possible that due to the train-test split, the data objects in the training data follow different rules than the objects in the test data (meaning the distribution of the feature values or the relationship between features and class labels can be different between training and test data). As a result, the accuracy on the test data can be 0% in the worst case, even if it is very high on the training data. However, one should avoid train-test splits that alter the distributions of features and class labels. One counter-measure is to stratify the split with regard to the class labels, i.e. make sure the proportions of the class labels are equal in training and test data. Also, reducing overfitting of the tree to the training data might improve performance on the test data.

Frage 6 - Conditional Risk (1 Punkt) [ID: 1051790]

You're almost done studying for the Data Science I exam. Only 10 hours of preparation time remain, but you are still not sure if you should take the exam. Should you fail, you would have to retake the exam later, with a preparation effort of another 40 hours. You're 80% sure you will pass the exam on the first try and 100% sure to pass it at the second try. How high is the conditional risk (expressed in hours) of continuing preparation and taking the exam?

Der Wert muss zwischen 18 und 18 liegen

80 % * 10 h + (100 % - 80 %) * (10 h + 40 h) = 18 h or

100 % * 10 h + (100 % - 80 %) * 40 h = 18 h

The preparation effort for the first exam is due in any case if you want to take it, no matter if your pass or fail. The second exam only needs to be prepared if the first one was not passed.

Frage 7 - Ensembles (1 Punkt) [ID: 1063773]

Suppose we want to train an ensemble model for a regression dataset.

Bagging *(0.34 Punkte)* ensembles take the unweighted mean over their individual regressors. In contrast,

boosting *(0.33 Punkte)* ensembles take a weighted mean. Finally, stacking *(0.33 Punkte)* ensembles

use a meta-model to weight their individual regressors.

See the lecture.

Frage 8 - Training, Validation and Test (1 Punkt) [ID: 1051791]

Assign the different data splits to their main purpose!

Training data	passt zu	Building a model	*(0.34 Punkte)*
Validation data	passt zu	Comparing models and hyperparameter settings	*(0.33 Punkte)*
Test data	passt zu	Evaluating the final model configuration	*(0.33 Punkte)*

See the lecture. After selecting a model and hyperparameter configuration, validation data can be used for training as well. After determining the quality of the finally chosen configuration, test data can be used for training as well.

Frage 9 - Bias and Variance - Train/Test Performance (1 Punkt) [ID: 1051792]

Assume a model has

a) a training accuracy clearly below 100%, e.g., 60%

b) a large gap between training and test accuracy

What is the name of the underlying phenomena?

- ○ a) Bias, b) Bias
  (0 Punkte)
- ◉ a) Bias, b) Variance
  (1 Punkt)
- ○ a) Variance, b) Bias
  (0 Punkte)
- ○ a) Variance, b) Variance
  (0 Punkte)

If the model already is rather inaccurate on the training data, it might not be complex enough to capture the dependencies in the data. This means it has a large bias. Further reasons for low training performance are bugs in the training process or insufficiently discriminating features.

If there is a large gap in prediction performance between training and test, the model usually has overfit to the training data, causing non-generalizable predictions. This means the model is highly sensitive to input data and has a high variance.

Frage 10 - Bias and Variance - Models (1 Punkt) [ID: 1063770]

Match the following predictions models to either "high bias" or "high variance", depending on which danger is higher!

Linear regression	passt zu	High bias	*(0.25 Punkte)*
k-NN with low k	passt zu	High variance	*(0.25 Punkte)*
Unpruned decision tree	passt zu	High variance	*(0.25 Punkte)*
Naive Bayes	passt zu	High bias	*(0.25 Punkte)*

Simple linear models might not be strong enough to fit complex dependencies in the data, thus they are biased. In contrast, if using higher order polynomials and interaction terms, the danger of overfitting (high variance) grows.

k-NN with low k strongly depends on a few points in the query's neighborhood and thus is subject to high variance. By increasing k, variance decreases, but bias increases, as more and more points with strongly different feature values are considered.

Unpruned decision trees overfit the data and exhibit high variance. Pruning might reduce variance, but introduce bias (it the tree becomes too weak to fit the data).

Naive Bayes, as linear models, might underfit the data. More complex Bayesian models could reduce bias, but might introduce more variance (in particular, if there is not enough data to estimate multivariate distributions properly).

Frage 11 - Baseline Performance of Evaluation Measures (1 Punkt) [ID: 1051793]

Assuming two classes are present in the data, for which evaluation measure can you easily get a value of 1 (= 100%) by simply always predicting the "positive" class?

- ○ Accuracy
  (0 Punkte)
- ○ Kappa coefficient
  (0 Punkte)
- ○ Precision
  (0 Punkte)
- ◉ Recall
  (1 Punkt)

Recall is defined as $\frac{TP}{TP+FN}$. By always guessing "positive", you can't produce false negatives. You still get false positives, but they don't count towards the recall, in contrast to the other evaluation measures mentioned in the task.

Frage 12 - Cohen's Kappa Coefficient (1 Punkt) [ID: 1051794]

Assume the following values of a contingency table to evaluate classification:

TP = 60, FP = 10, FN = 10, TN = 20

Calculate Cohen's kappa coefficient. Round your solution to two decimal places and use a point (.) as decimal separator.

Der Wert muss zwischen 0.52 und 0.52 liegen

The classifier predicts class "positive" with 70% (TP + FP) and class negative with 30% (TN + FN). In the ground truth, 70 data objects belong to "positive" (TP + FN) and 30 data objects belong to "negative" (TN + FP). If you guess class membership with the same proportions as the classifier, you classify 70% * 70 + 30% * 30 = 58 data objects correctly (for the 70 "positive" objects, your guess is correct in 70% of the cases, and for the 30 "negative" objects, your guess is correct in 30% of the cases). The classifier has classified 80 data objects correctly (TP + TN). These numbers yield a kappa coefficient of $\frac{80-58}{100-58} = 0.52$. (You can also calculate with the frequencies, i.e., $\frac{0.8-0.58}{1-0.58} = 0.52$)

Note that an even stronger baseline than "guessing with same proportions" here is "always guessing positive", yielding an accuracy of 70% instead of 58%.

Frage 13 - ROC and Lift (1 Punkt) [ID: 1051795]

Assign the evaluation metrics to the plots they appear in. (Hint: each plot shows two metrics.)

ROC plot	passt zu	True positive rate (recall)	*(0.25 Punkte)*
ROC plot	passt zu	False positive rate	*(0.25 Punkte)*
Lift plot	passt zu	Number of true positives	*(0.25 Punkte)*
Lift plot	passt zu	Number of data objects	*(0.25 Punkte)*

See the lecture.