Classification Metrics

Applied ML in Engineering - Exercise 09

TU Berlin, Summer Term 2023

Prof. Dr.-Ing. Merten Stender – merten.stender@tu-berlin.de

Students are asked to implement different classification metrics as well as summary statistics for competing metrics, such as the area under curve (AUC) for the receiver-operator characteristic *or* the precision-recall curve. Students can choose between two options:

1. Balanced data treatment: Data set `dataset_balanced.txt` from previous exercise. Metrics: ROC and corresponding AUC
2. Imbalanced data treatment: Data set `dataset_imbalanced.txt` from homework no 1. Metrics: PRC and corresponding AUC.

Ground truth labels are always given in the second-last column, and the probability predictions by a logistic regression model are always given in the last column.

Problem 1: 20min

Assume a binary classification task and binary vectors `y` (ground truth) and `y_pred` (prediction) both carrying ones (positive class) and zeros (negative class).

(a) Implement subroutines for computing the entries of the confusion matrix TP, TN, FP, FN

(b) Make use of the `numpy` function `where()` for writing short and comprehensive conditional tests

Problem 2 (balanced data): 45min

Assume binary ground truth labels and probability predictions, i.e. continuous values $\in [0, 1]$.

(a) Implement the true and false positive rate in `fpr()`, `tpr()` assuming binary labels and predictions

(b) Implement the ROC curve computation in `roc(y, y_pred)` that is returning the `fprate` and `tprate`. You may need to implement a helper function to binarize continuous predictions according to the probability threshold `thresh`.

(c) Plot the ROC for the exemplary data provided in `dataset_balanced.txt` and compare against Figure 1

(d) Compute the area under the curve (AUC) for the ROC return values. Use `numpy.trapz` to compute the integral.

Problem 2 (imbalanced data): 45min

Assume binary ground truth labels and probability predictions, i.e. continuous values $\in [0, 1]$.

(a) Implement the precision and recall in `precision()`, `recall()` assuming binary labels and predictions. You may need to implement a helper function to binarize continuous predictions according to the probability threshold `thresh`.

(b) Implement the precision-recall curve computation in `prc(y, y_pred)` that is returning the `precision_` and `recall_`.

(c) Plot the PRC for the exemplary data provided in `dataset_imbalanced.txt` and compare against Figure 1

(d) Compute the area under the curve (AUC) for the PRC return values. Use `numpy.trapz` to compute the integral.

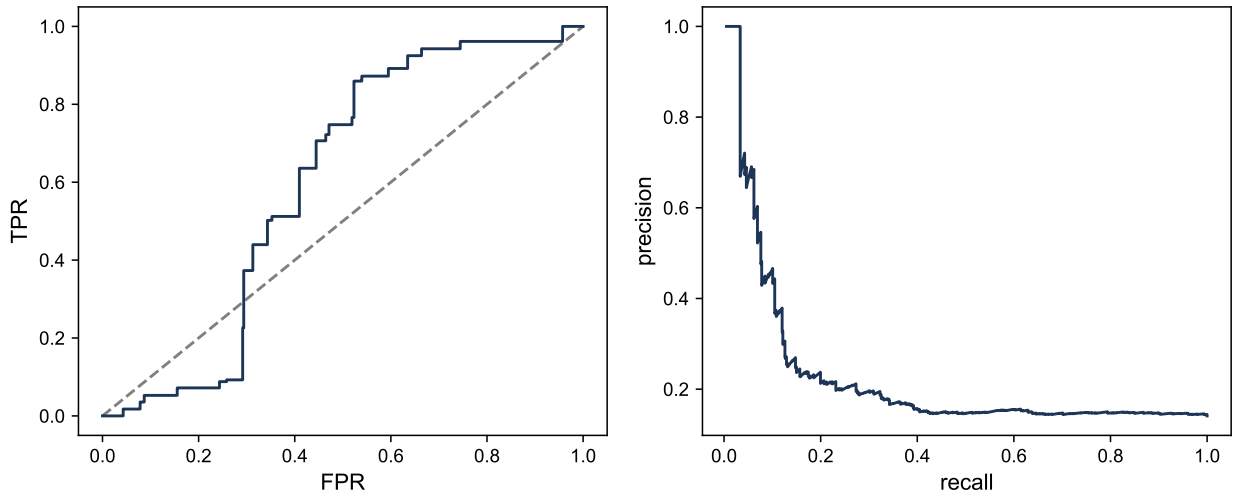

Figure 1: ROC (left) and PRC (right) for the exemplary data sets (balanced, imbalanced). ROC-AUC$= 0.598$, PRC-AUC$= 0.085$

Problem 3: 25min

Choose one dataset and try to find a best-performing model in terms of AUC using methods from sklearn. Try to be better than the default logistic regression model used for generating the predictions supplied in the data files.