# Performance metrics – exercises

The description of the used performance metrics can be found in chapter 3 (for the classification metrics) and chapter 2 (for the regression metrics) of the book ‘Hands-On Machine Learning with Scikit-Learn and TensorFlow’.

## Exercise 1 – performance metrics for binary classification

We have a dataset with 1158 patients, of who 105 patients have a certain disease that we are trying to predict. This ratio is typical for the disease, as it has a prevalence of approximately 10%.

1. A trained model achieves an accuracy of 0.91. What can you conclude about the usefulness of the model based on only the accuracy?

**This would suggest the model determines the condition correctly circa 91% of the times. This is quite problematic as the disease occurs realatively often (about 10% of the times) and out of these a large chunk will be missed.**

1. Another model is trained, for which the confusion matrix looks as follows:

**Predicted**

**Actual**

|  |  |  |
| --- | --- | --- |
|  | Positive | Negative |
| Positive | 91 | 14 |
| Negative | 33 | 1020 |

1. What are the false positives in this case? And what are the false negatives?

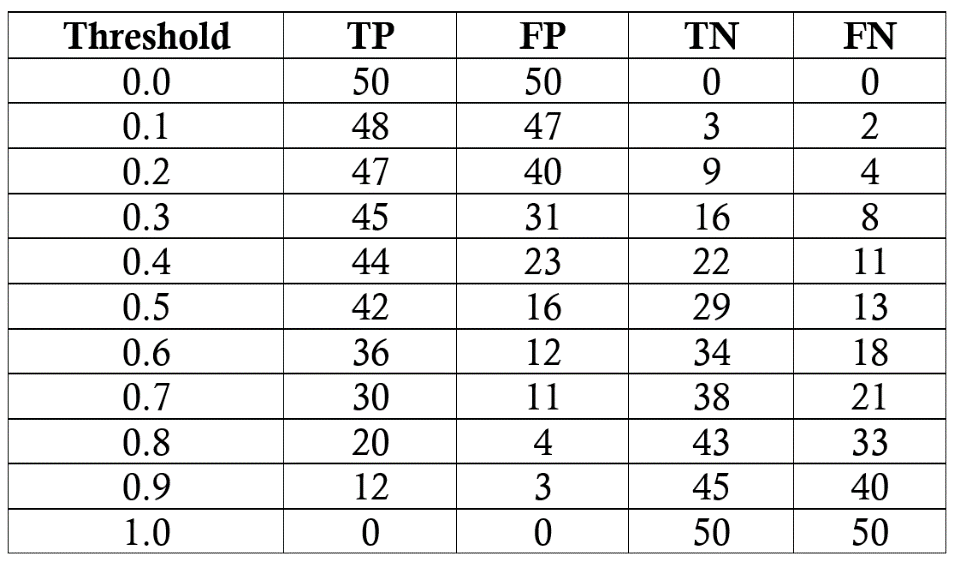
**False positives: 33, False negatives: 14**

1. Compute the accuracy for this model.
2. Compute the precision for this model. Describe what precision means in this case.
3. Compute the recall for this model and describe what recall means in this case.
4. Compute the F1-score.
5. Describe what is meant by the ‘precision/recall tradeoff’.
6. Suppose people that have the disease experience just some minor discomforts, while the follow-up examination of the disease is quite expensive. How would you define a good model in this case in terms of precision and recall? Is the F1-score a useful metric in this case?
7. In this example, the model was trained and tested on the same dataset. Why is this in general a bad practice? How can you solve this?

## Exercise 2 – The ROC-curve

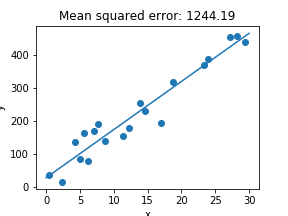
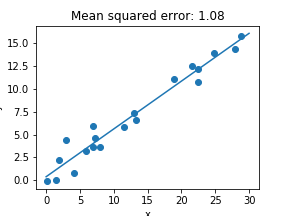
We’re considering a different disease now. This disease is present in 50% of the general population, and we have a dataset containing 100 patients. We will assume a black box model which has as outcome a score between 0 and 1. A threshold can be set to decide when a patient is labelled as positive. For example, at a threshold of 0.6 a patient that receives a score of 0.53 is predicted negative (or 0). At different thresholds the true positives, true negatives, false positives and false negatives are computed. The results are in the table below. In a ROC-curve, the true positive rate is plotted against the false positive rate for different thresholds.

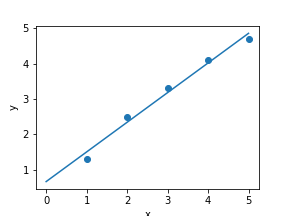
1. Create the ROC-curve based on the data in the table below.
2. When do you use ROC instead of a precision/recall curve?



## Exercise 3 – performance metrics for regression

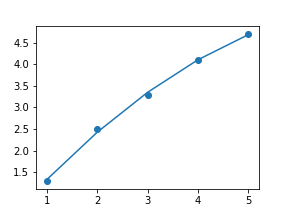
1. Typically Python works with Mean Squared Error (MSE) instead of RMSE. Note that RMSE is just the Root of the MSE. In the figures below, linear regression was done on different data sets and the mean squared error was computed. Note the scale on the y-axis. Can you say anything about whether one regression is better than the other? Why (not)?



1. On the following values of x and y, linear regression is performed. The corresponding prediction function is: *y\_pred = 0.84x + 0.66*. Compute the predicted values for y at the given x-values, and compute the RMSE.

|  |  |  |
| --- | --- | --- |
| x | y | y\_pred |
| 1 | 1.3 |  |
| 2 | 2.5 |  |
| 3 | 3.3 |  |
| 4 | 4.1 |  |
| 5 | 4.7 |  |

1. A polynomial regression is done on the data as shown in the figure below, resulting in a RMSE of 0.04. What can you say about the performance of this model compared to the linear model in c?

****