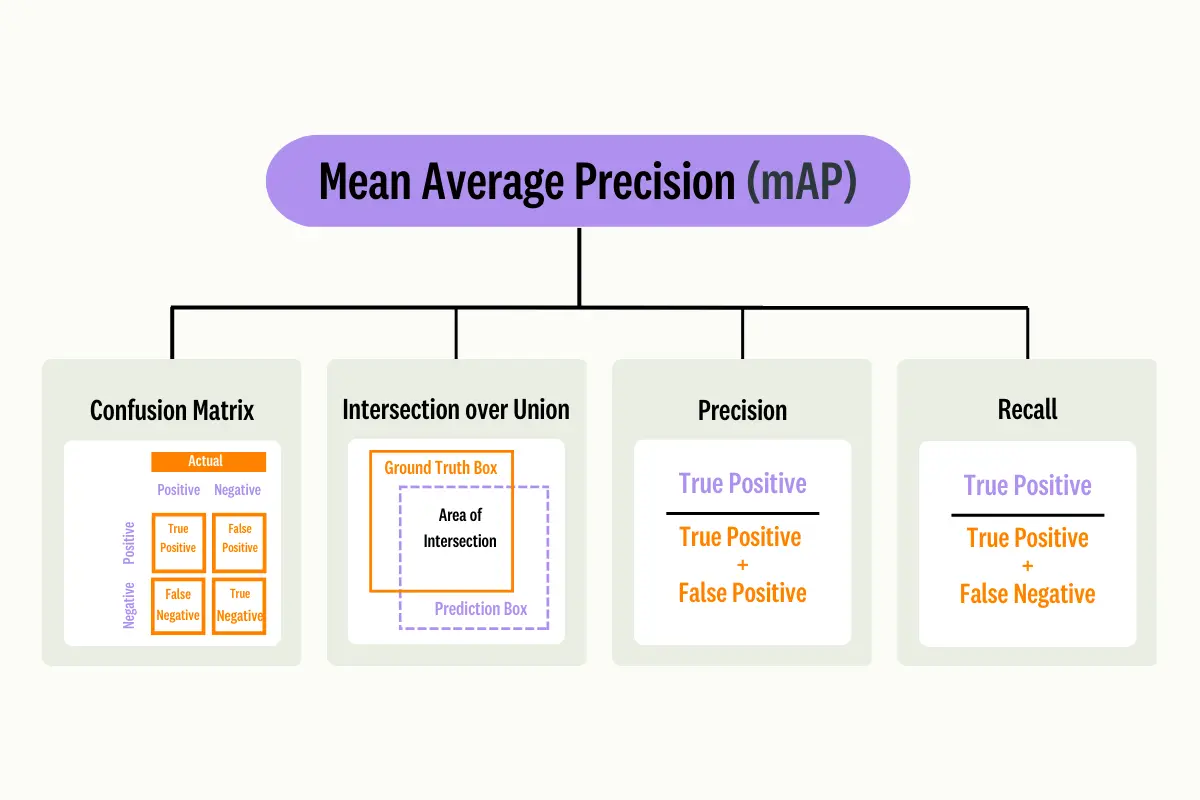
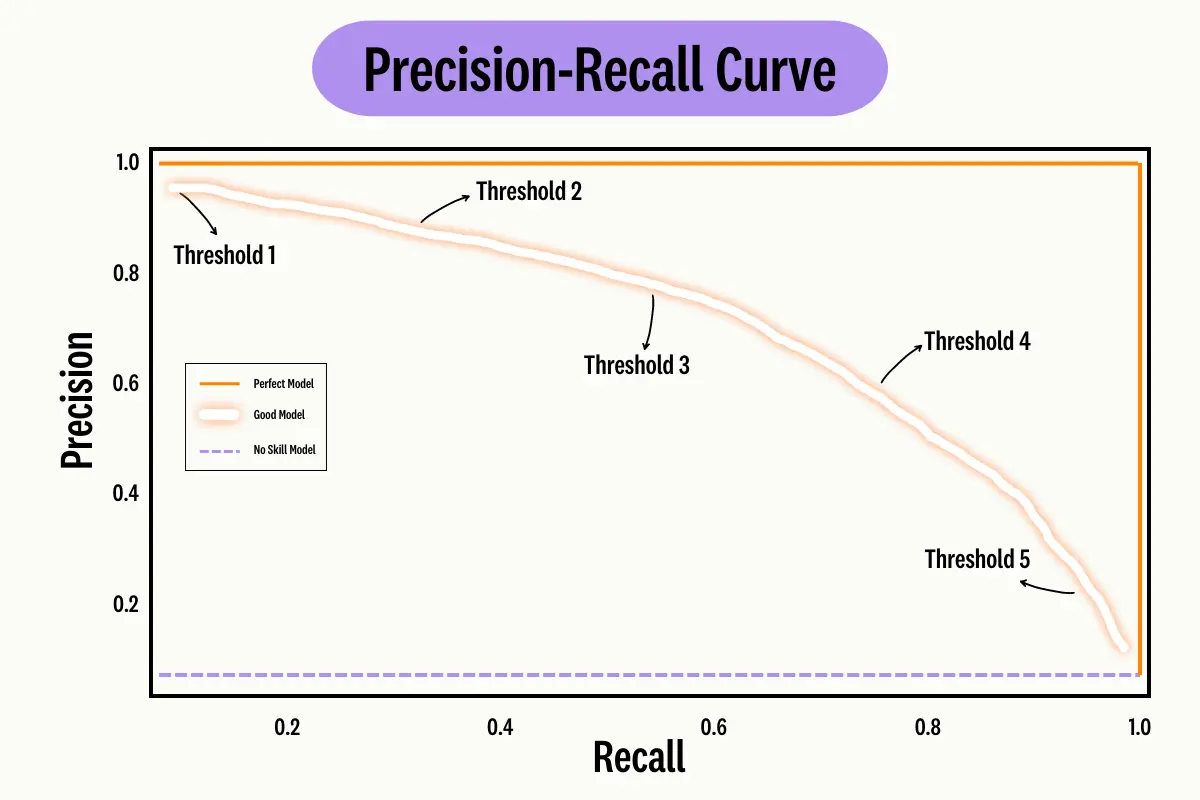
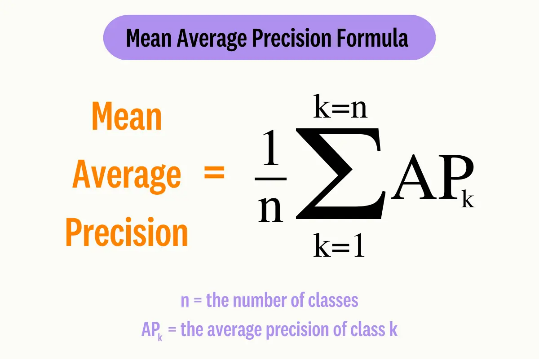
Write identification

Codebook generation For computing the residuals (xt − µk ), we need a codebook to generate the dictionary D. To do so, a representative number of descriptors needs to be selected. Therefore, use a random selection of about 500 000 descriptors from the training features. Compute a codebook (a. k. a. vocabulary/background model) with the help of k-means (k = 100). For this operation you can use sklearn’s MiniBatchKMeans.





*For example, if a model has a high recall value but a low precision value, it means that the model is classifying as many negative samples as it is positive samples. If a model has a high precision value, but a low recall value, it means that the model has the ability to classify samples as positive, but only some.*

**

*The concept of precision at K used in the calculation of mAP (AP @ K) stands for the Mean Average Precision at K. It is used to evaluate if the predicted items are relevant and if the most relevant items are at the top. The number of correctly labeled predicted labels is calculated, where K represents the top K labels that are considered.*

*Therefore, the Average Precision at K is the sum of the precision at K of the values of K divided by the total number of relevant items in the top K results. If we were to calculate the mean average precision at K, we measure the Average Precision at K averaged over all queries (entire dataset).*

A diagram of a graph

Description automatically generated with medium confidence

b) VLAD encoding With the help of the computed dictionary, we can now compute the residuals (see above). They are multiplied with the indicator/association function αk(xi), which returns 1 for the nearest cluster center and 0 otherwise. Therefore, you can use cv2.BFMatcher and knnMatch. Suggestion: compute first the full association matrix (i.e. a vector for each descriptor. Each element of the vector represents the association to a cluster. That means, set all elements of the vector to 0 except the nearest cluster center). This can then be used to obtain for each cluster only those descriptors that have this cluster assigned. Then, compute the residuals with these descriptors and aggregate them to one global descriptor. Compute the pairwise distances between all encodings and evaluate the performance in terms of mean average precision (mAP).

c) VLAD normalization In order to improve the representational power of VLAD encoding, we should account for visual burstiness, i.e. frequent local descriptors that dominate the overall similarity. This can be achieved by using power normalization: ξ ′ i = sign(ξi) p |ξi |, ∀{ξi} ∈ ξ, where sign is +1 if ξi ≥ 0 and −1 otherwise. Don’t forget to ℓ 2 normalize, i. e., divide by the ℓ 2 norm, the representation again afterwards. Compute the pairwise distances between all normalized encodings and evaluate the performance in terms of mean average precision (mAP) and compare with the result before

d) Exemplar classification The last step of the pipeline is the proposed exemplar classification. Therefore, you have to compute an individual SVM for each global representation (encoding) of the test set using this encoding as positive and all the encodings of the training set as negatives. (sklearn’s LinearSVC), a good C value is 1000 (actually this value should be cross-validated with a different validation set), also set class weight to balanced. Afterwards, ℓ 2 normalize the weight vector (coef parameter of the SVM) and use it as your new global descriptor. Compute the pairwise distances between all newly created descriptors and evaluate the performance in terms of mean average precision (mAP) and compare with the result before.