INFO371 Problem Set: Non-parametric machine learning models

knn, decision trees, and random forests

Due 10/22/2023, 11:59pm

Instructions

1. Please write clearly! Answer each question in a way that if the code chunks are removed from your document, the result is still readable!

2. Discussing the solutions and getting help is OK, but you have to solve the problem on your own. Do not copy-paste from others. Do not solve the problems with generative AI.

1 A simple classification task (5 pts)

Here we use a skin tone dataset *skin-nonskin.csv*. It contains a large number of colors (as R, G, B), and a label for skin/non-skin tone ("1" = skin, "2" = non-skin). Here is an example of colors and the corresponding labels:

235,179,162: 1	222,150,102: 1	87,77,127: 2	22,71,68: 2
229,180,165: 1	248,183,143: 1	110,34,46: 2	2,2,2: 2
216,151,113: 1	223,169,133: 1	245,68,94: 2	10,30,29: 2
219,161,124: 1	199,133,72: 1	23,59,57: 2	3,18,21: 2
223,169,133: 1	221,165,128: 1	163,197,199: 2	164,199,201: 2
129,89,81: 1	253,185,148: 1	136,176,176: 2	14,50,50: 2
192,117,75: 1	154,101,49: 1	162,197,199: 2	133,181,181: 2
182,117,61: 1	255,217,208: 1	105,85,180: 2	3,14,16: 2
163,104,46: 1	145,98,56: 1	118,168,169: 2	0,4,4: 2
127,89,76: 1	216,144,96: 1	92,141,138: 2	36,90,90: 2

We recommend the sklearn library for the following tasks.

Explore and prepare the data

1.1 (1pt) Load the data. Find:

(a) the number of rows and columns

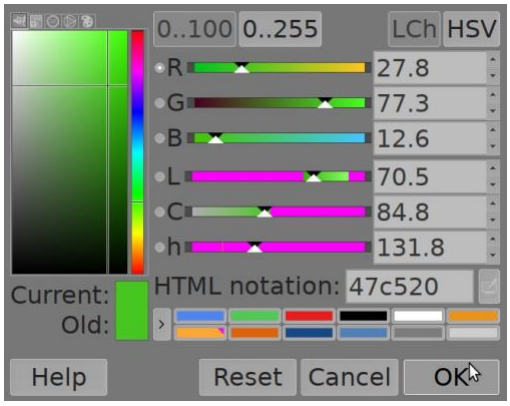
(b) print a few lines of data

(c) does it contain any missing values?

(d) what are maximum and minimum values for R, G, B?

(e) what are the possible labels?

2. (1pt) Note that the feature space here is about the same as "color space", you have probably seen color selectors that work on such a color space. Here is the color selector of gimp:



(a) What is the dimensionality of the feature space?

(b) In this feature space, does the class of feasible skin tones have a linear or non-linear boundary?

(c) Given the R,G,B values, is there any uncertainty if the given tone is a possible skin tone?

3. (1pt) What is the accuracy of a naive classifier that categorizes all samples as the majority class?

4. (1pt) Create the feature space X (the R, G, B values) and the target variable y (the labels). Split both X and y into training and validation sets (80% for training, 20% for testing).

1.2 knn (1pt)

1. Compute accuracy on both training and validation data by using knn.

2. How well does knn perform compared to the naive model above and why?

2 Decision trees (6 pts)

Now it is time for decision trees. Download loan_approval_dataset.csv. Create the feature space X and the target variable y. (Hint: Remove the loan_id column and convert features that have string values to categorical entries, e.g.,

var0 = {"education": {"Graduate":0, "NotGraduate":1}}

df = df.replace(var0)

Split both X and y into training and validation sets (80% for training, 20% for testing).

2.1 Maximum depth in decision tress (3pt)

The task here is to first tune two parameters independently, and afterward tune both together. As above, compute both training and validation accuracy.

Some of the tasks may be rather slow. To speed it up, you may not want to test all possible parameter values. For instance, you may go from depth 1 to 100, but only test every 5th or 10th value.

1. (0.5pt) What kind of decision boundary do you expect to see based on trees?
2. (0.5pt) Explain what the maximum depth parameter does. Do large or small values for maximum depth cause overfitting? (Hint: Check out sklearn's documentation.)
3. (1pt) Run a series of decision tree models of different maximum depth in a loop. Start with a small depth, and increase it into the overfitting territory so that the model starts overfitting. At each iteration, store both validation and training accuracy.

   Make a plot where you show how both training and validation accuracy depend on maximum depth. Try to make the graph so that the differences are easily visible. (Hint: Try plotting log(1−accuracy) instead of just accuracy.)
4. (0.5pt) What is the best validation accuracy that you get? What is the corresponding maximum depth?
5. (0.5pt) Discuss your findings: where does the model start overfitting? What is the optimal depth?

2.2 Minimum sample size to split (3pt)

Next, let's repeat 1.3 with "min_sample_split":

1. (0.5pt) Explain what the min_sample_split parameter does. Do large or small values lead to overfitting?
2. (1pt) Run a series of decision tree models with different min_sample_split values in a loop. Try to cover both underfitting and overfitting. Each time store both validation and training accuracy. Make a plot where you show how both training and validation accuracy depend on the parameter.
3. (0.5pt) What is the best validation accuracy you get? What is the corresponding min_sample_split?
4. (0.5pt) Discuss your findings: where does the model start overfitting?

1. (0.5pt) Write a double loop over both parameters. Try to pick a number of values not too different from what you found above when analyzing those individually. Store the best result and the respective parameters.

3 Random forests (4 pts)

Now it is time to use random forests. Check out sklearn's RandomForestClassifier.

1. (1pt) Explain what the n_estimators parameter does.
2. (1pt) Run a series of random forest classifiers testing what is the best number of estimators. This may be slow, so you may want to skip quite a few potential values.
3. (1pt) Make a plot where you show how both training and validation accuracy depend on the parameter. What is the best validation accuracy you get?
4. (1pt) Where does the model start overfitting, please explain?