HW3 report

1 KNN and Model Selection (k) (programming)

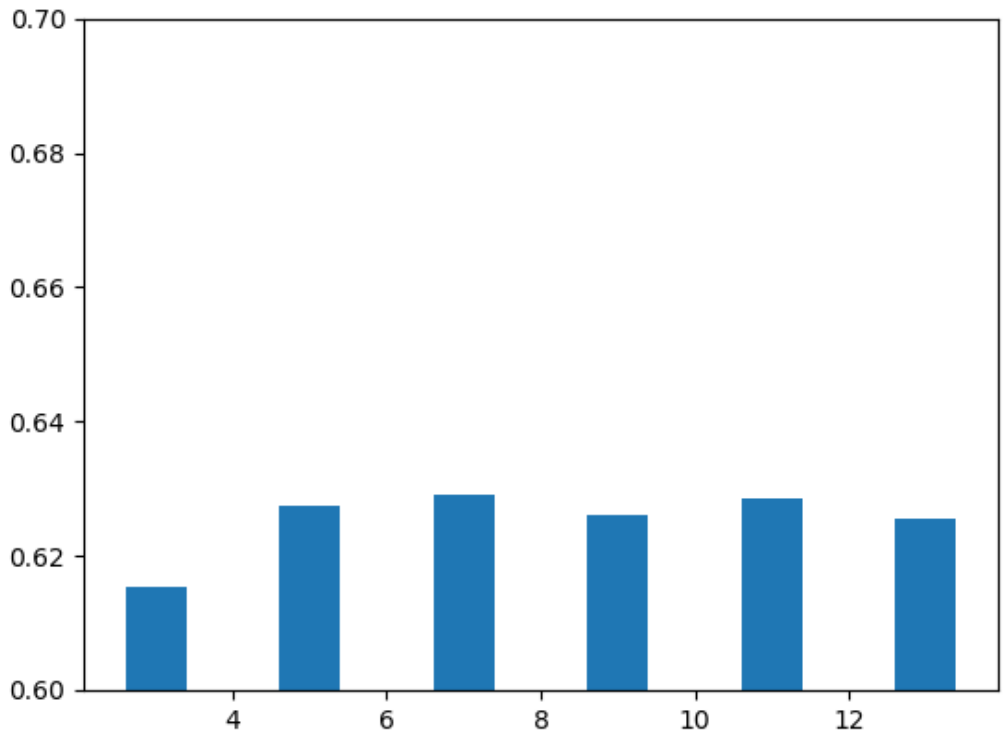
Validation Accuracy

k	Accuracy
3	0.6154999999999999
5	0.6275000000000001
7	0.629
9	0.626
11	0.6285
13	0.6255000000000001

Best k: 7.

When k is too small, prediction are more prone to outliers and tend to be unstable, while when k is too large, the model will take training example that may not be close to the test example into consideration.

k v.s. accuracy



2 Support Vector Machines with Scikit-Learn and preprocessing

Results

Kernel	Params	CV train accuracy	CV test accuracy
rbf	C=1, degree=1	0.8286878985264713	0.8294328158389331
rbf	C=1, degree=3	0.8286878985264713	0.8294328158389331
rbf	C=1, degree=5	0.8286878985264713	0.8294328158389331
linear	C=1, degree=1	0.8535671069230967	0.8532993486264514
linear	C=1, degree=3	0.8535671069230967	0.8532993486264514
linear	C=1, degree=5	0.8535671069230967	0.8532993486264514
linear	C=1, degree=7	0.8535671069230967	0.8532993486264514
poly	C=1, degree=1	0.834901264128112	0.8357148374140727
poly	C=1, degree=3	0.8351055156492709	0.8357405834041348
poly	C=1, degree=5	0.8369746745277756	0.8370021369171751
poly	C=1, degree=7	0.8406151575225492	0.8405808295357997
sigmoid	C=1, degree=3	0.7158878504672898	0.7159187456553641
sigmoid	C=1, degree=5	0.7158878504672898	0.7159187456553641
sigmoid	C=1, degree=7	0.7158878504672898	0.7159187456553641

Best performing model:

- kernel: linear
- C = 1
- degree = 1

How I preprocessed the data

First I one-hot encoded the columns `['workclass', 'education', 'marital-status', 'occupation', 'relationship', 'race', 'sex', 'native-country']` Then I normalized the continuous columns `['age', 'fnlwgt', 'capital-gain', 'capital-loss', 'hours-per-week']` Finally I stripped the column `label` and mapped '<=50K' to 0 and '>50K' to 1

How I chose the parameters

I tested all the available kernels, including rbf, linear, poly, and sigmoid, and then I took the best performing model, which was the linear model.

3 Sample QA Questions

Question 1. Support Vector Machine

(a) False. C determines how much slack we allow. If we allow a large amount of slack, we will have a large number of support vectors. If we allow very little slack, we will have very few support vectors.

(b) 2

(c) 2, 1, 3