Assignment 5: K-Nearest Neighbor Classifier and Support Vector Machines

UVA CS4774

Machine Learning Foundation, Deep Learning and Good Uses

May 1, 2022

Owen Richards

1. KNN and Model Selection (k) (programming):

Task 1:

In the tables below it reports the cross-validation accuracy for all values of k as well as the best k, for all three types of input features. All the features used a 4-fold cross validation strategy for selecting the best k for KNN classification. Some values of k work better than others and this can be down to model complexity. The larger the k is, the more likely k is to underfit since it is making the model more generic. When k is large, the model may take training examples that may not be close to the test examples. However, the smaller the value of k is the model will be affected more easily to noise and could overfit the training set. Therefore, the different features have varying best k values since it depends on the complexity of the model.

Bow Representation

Besk_k value: 13, Test Accuracy: 0.65666666666666

K	Accuracy
3	0.5714285714285714
5	0.5714285714285714
7	0.585
9	0.5878571428571429
11	0.5921428571428572
13	0.5985714285714285

TFIDF Representation

Best_k value: 13, Test accuracy 0.8283333333334

K	Accuracy
3	0.6557142857142857
5	0. 6557142857142857
7	0.6678571428571428
9	0.6749999999999999
11	0.6914285714285715
13	0.706428561428514

BERT Representation

Besk_k value: 7, Test Accuracy 0.799

K	Accuracy
3	0.6578571428571429
5	0.6664285714285714
7	0.6864285714285715
9	0.6792857142857143
11	0.6764285714285715
13	0.6814285714285715

Task 2:

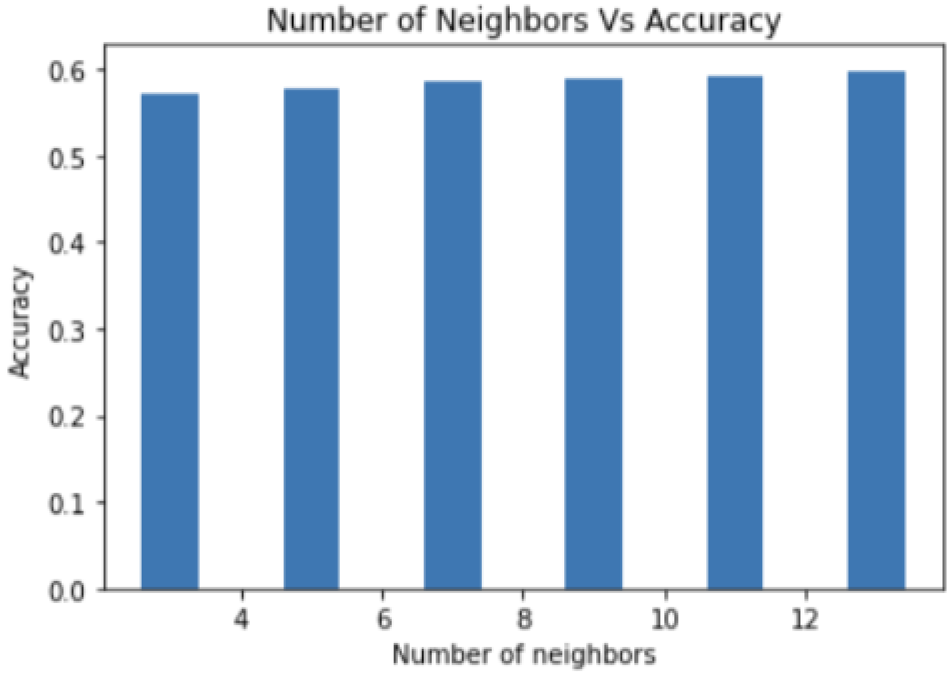

Figure 1: BOW Representation Bar Graph

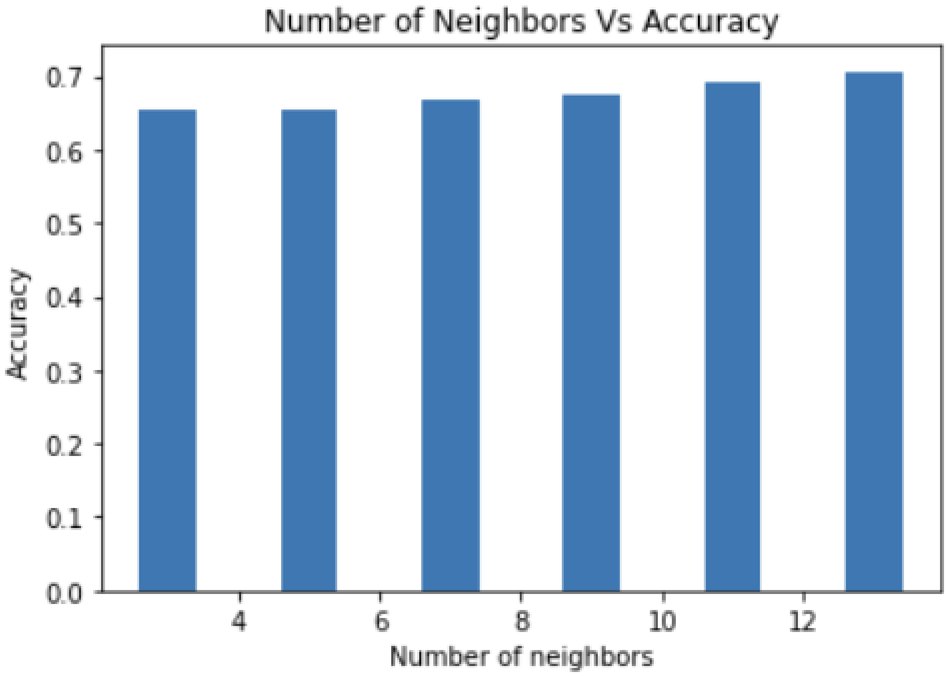

Figure 2: TFIDF Representation Bar Graph

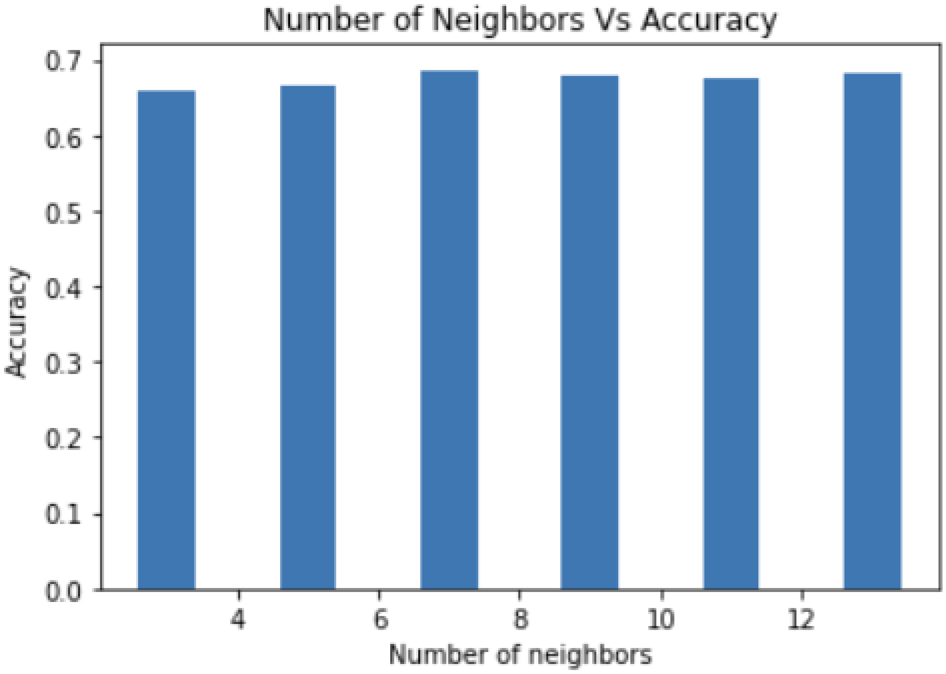

Figure 3: BERT Representation Bar Graph

2. Support Vector Machines with Scikit-Learn and preprocessing

Below in Figure 4 is a complete table of all the hyperparameters used and the classification results. The best performing kernel was with “rbf”, with a C value of 13. The train accuracy was 0.8629 and the CV test accuracy was 0.8551. I expect that with a larger C value the train accuracy and test accuracy would increase more; however, the time to run the program would also increase.

Kernel	Kernel Parameters	CV Train Accuracy	CV Test Accuracy
rbf	C : 1	0.8541	0.8519
rbf	C : 3	0.8563	0.8529
rbf	C : 5	0.8586	0.8534
rbf	C : 7	0.8600	0.8540
rbf	C : 9	0.8609	0.8540
rbf	C : 13	0.8629	0.8551
poly	C : 1, Degree : 1	0.8504	0.8502
poly	C : 1, Degree : 3	0.7834	0.7834
poly	C : 1, Degree : 5	0.7669	0.7668
poly	C : 1, Degree : 7	0.7654	0.7654
poly	C : 3, Degree : 1	0.8518	0.8516
poly	C : 3, Degree : 3	0.8177	0.8170
poly	C : 3, Degree : 5	0.7714	0.7712
poly	C : 3, Degree : 7	0.7655	0.7653
linear	C : 1,	0.8533	0.8521
linear	C : 3	0.8533	0.8520
linear	C : 5	0.8533	0.8523
linear	C : 7	0.8533	0.8522

Figure 4: CV Classification Accuracy Results for All Hyperparameters

I did the preprocessing in three main steps using some built-in functions such as the sklearn OneHotEncoder, LabelEncoder and the StandardScaler. The three main steps to do was to encode the categorical features, then to normalize the continuous columns and then strip the column label and map ‘<=50k’ to 0 and ‘>50k’ to 1. OneHotEncoder was used to encode the categorical features into one-hot numeric array. Then StandardScaler was used to normalize the continuous attributes. Finally, I used the sklearn LabelEncoder to map ‘<=50k’ to 0 and ‘>50k’ to 1. This seemed like the best way to preprocess the data so that I could get the best accuracy.

For the SVM hyperparameters I first tried the different SVM kernels which as the basic linear kernel, the polynomial kernel and the RBF kernel. I noticed the general trend for these kernels was that with a larger C value the better the accuracy became. I found that rbf had the greatest improvement with each increase in the C value. However, the larger the C value was the longer it took so I ended up having my best accuracy with kernel rbf and a C value of 13.