HW2_Jingjie Zhou

1.

(a)

for K=1, 5, 11, 21, 41, 61, 81, 101, 201, 401

test accuracies are

[0.7522816166883963, 0.7548891786179922, 0.7648848326814428, 0.7466318991742721, 0.7522816166883963, 0.7375054324206867, 0.7266405910473707, 0.7288135593220338, 0.7314211212516297, 0.7196870925684485]

(b)

for K=1, 5, 11, 21, 41, 61, 81, 101, 201, 401 with z-score normalization

test accuracies are

[0.8231203824424164, 0.8322468491960018, 0.8748370273794003, 0.8709256844850065, 0.8704910908300739, 0.8700564971751412, 0.8696219035202086, 0.8639721860060843, 0.8461538461538461, 0.8144285093437635]

(c)

['t1', 'spam', 'spam', 'spam', 'spam', 'spam', 'no', 'no', 'no', 'no', 'no']
['t2', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'no', 'no', 'no']
['t3', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t4', 'spam', 'spam', 'spam', 'spam', 'no', 'no', 'spam', 'spam', 'spam', 'spam']
['t5', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t6', 'spam', 'spam', 'spam', 'no', 'no', 'spam', 'spam', 'spam', 'spam', 'spam']
['t7', 'spam', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no']
['t8', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t9', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t10', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t11', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t12', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t13', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'no', 'no', 'no', 'no']
['t14', 'no', 'spam', 'spam', 'spam', 'no', 'no', 'no', 'no', 'no', 'no']
['t15', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t16', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t17', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t18', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'no', 'no', 'no']
['t19', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t20', 'no', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t21', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t22', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'no', 'no', 'no', 'no']
['t23', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t24', 'no', 'no', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t25', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t26', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t27', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t28', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t29', 'spam', 'spam', 'spam', 'no', 'spam', 'spam', 'spam', 'spam', 'no', 'no']

['t30', 'spam', 'spam', 'spam', 'spam', 'no', 'no', 'no', 'no', 'no', 'no']
['t31', 'spam', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no']
['t32', 'spam', 'spam', 'spam', 'spam', 'no', 'spam', 'spam', 'spam', 'no', 'no']
['t33', 'spam', 'spam', 'spam', 'spam', 'no', 'no', 'no', 'no', 'no', 'no']
['t34', 'spam', 'spam', 'no', 'spam', 'no', 'no', 'no', 'no', 'no', 'no']
['t35', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t36', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t37', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t38', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t39', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t40', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no']
['t41', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no']
['t42', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'no', 'no']
['t43', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no']
['t44', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no']
['t45', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t46', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t47', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t48', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t49', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']
['t50', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam', 'spam']

(d)
test accuracies increase with z-score normalization applied to the features.
z-score can avoid some features that have large scale and some don’t, which large scale feature will affect the result more.

(e)using 10-fold cross-validation to choose the right k. Separate data of ‘spam train.csv’ 10 splits. Each time, using 9 of them as train data while the rest as test dataset. Calculate the average test error. Choose k value with the smallest test error.

2.
Top 6,4 0.971
Education gain = 0.124
	1.High School 4,1 0.722
		Experience gain = 0.322
			1.’Less than 3’ 0,1 0-----L
			2.’3 to 10’ 0,2 0 ----------L
			3.’More than 10’ 1,1 1
				Career gain = 0.322
					1.Management 1,0 0------H
					2.Service 0,1,0 ----------L
	2.College 3,2 0.971
		Career gain = 0.42
			1.Management 2,0 0-------H
			2.Service 1,2 0.918

Experience gain = 0.42

1.Less than 3 -------L

2.3 to 10 ----------H

3.More than 10-----L

prune:

before pruning, the validation set has 1 correct and 2 wrong.

For high school-Experience-Career branch,

1. I try to prune career feature, make it ‘High’.the result stays the same, 1 correct 2 error. so I prune it.
2. I try to prune experience feature,make it ‘High’. the result stay the same, so I prune it.

For College-Career-Experience branch,

1. I try to prune experience, make it ‘Low’, the error decrease one, so I prune it.
2. I try to prune Career, make it ‘Low’, the error decrease one, so I prune it.

So, the tree only left one feature, which is Education and we can separate the data using just High school and College.

3.

PolyKernel:

exponent: 1

Correctly Classified Instances	712	84.1608 %
Incorrectly Classified Instances	134	15.8392 %

exponent: 2

Correctly Classified Instances	803	94.9173 %
Incorrectly Classified Instances	43	5.0827 %

exponent: 4

Correctly Classified Instances	788	93.1442 %
Incorrectly Classified Instances	58	6.8558 %

when the exponent is 1, the model do not perform well, it is because we do not actually make the data into high-dimensional space, so the data still hard to separate. But when we increase exponent, we make the dataset into high-dimensional space, and they became separatable.

RBFKernel:

gamma:0.01

Correctly Classified Instances	615	72.695 %
Incorrectly Classified Instances	231	27.305 %

gamma:1

Correctly Classified Instances	770	91.0165 %
Incorrectly Classified Instances	76	8.9835 %

when gamma is too small, the model is too constrain and can not get the dataset’s pattern, with gamma gets larger, the boundary gets wiggler and it is much easy to separate the dataset.

4.

$$K(x,z) = x_1 z_1 + x_1 e^{z_2} + z_1 e^{x_2} + e^{x_2 + z_2}$$
$$= \left(x_1 + e^{x_2}\right)\left(z_1 + e^{z_2}\right)$$
$$= f(x) f(z)$$

f(x) is the non-linear transformation of x , $\phi : x \to f(x) \in \mathbb{R}$

f(z) is the non-linear transformation of z, $\phi : z \to f(z) \in \mathbb{R}$

K(x,z) is the inner product of these two.