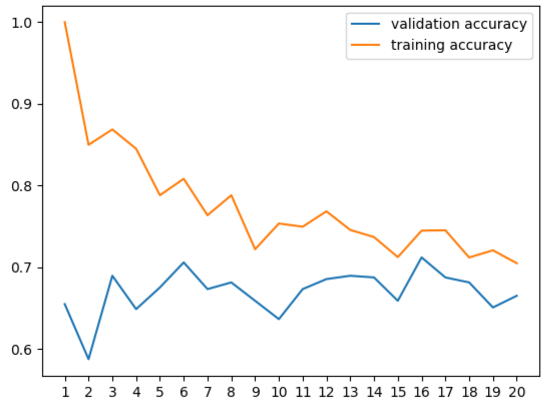
1.b.



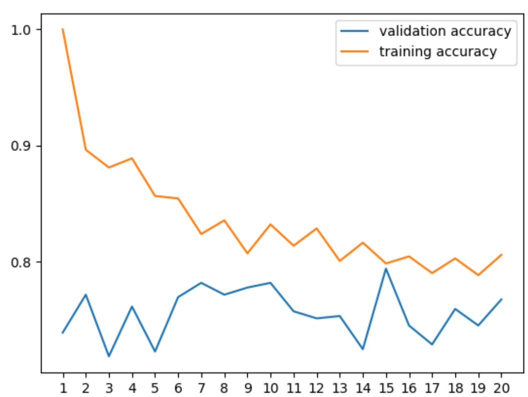

```
number of neighbors =   1
validation accuracy =   0.6551020408163265
training accuracy =   1.0
================================
number of neighbors =   2
validation accuracy =   0.5877551020408164
training accuracy =   0.8499562554680665
================================
number of neighbors =   3
validation accuracy =   0.689795918367347
training accuracy =   0.868766404199475
================================
number of neighbors =   4
validation accuracy =   0.6489795918367347
training accuracy =   0.8451443569553806
================================
number of neighbors =   5
validation accuracy =   0.6755102040816326
training accuracy =   0.7882764654418197
================================
number of neighbors =   6
validation accuracy =   0.7061224489795919
training accuracy =   0.8083989501312336
================================
number of neighbors =   7
validation accuracy =   0.673469387755102
training accuracy =   0.7637795275590551
================================
number of neighbors =   8
validation accuracy =   0.6816326530612244
training accuracy =   0.7882764654418197
================================
number of neighbors =   9
validation accuracy =   0.6591836734693878
training accuracy =   0.7222222222222222
================================
number of neighbors =   10
validation accuracy =   0.636734693877551
training accuracy =   0.7537182852143482
================================
number of neighbors =   11
validation accuracy =   0.673469387755102
training accuracy =   0.7497812773403325
================================
number of neighbors =   12
validation accuracy =   0.6857142857142857
training accuracy =   0.768591426071741
================================
number of neighbors =   13
validation accuracy =   0.689795918367347
training accuracy =   0.7458442694663167
================================
number of neighbors =   14
validation accuracy =   0.6877551020408164
training accuracy =   0.7370953630796151
================================
number of neighbors =   15
validation accuracy =   0.6591836734693878
training accuracy =   0.7125984251968503
================================
number of neighbors =   16
validation accuracy =   0.7122448979591837
training accuracy =   0.7449693788276466
================================
number of neighbors =   17
validation accuracy =   0.6877551020408164
training accuracy =   0.7454068241469817
================================
number of neighbors =   18
validation accuracy =   0.6816326530612244
training accuracy =   0.7121609798775154
================================
number of neighbors =   19
validation accuracy =   0.6510204081632653
training accuracy =   0.7209098862642169
================================
number of neighbors =   20
validation accuracy =   0.6653061224489796
training accuracy =   0.705161854768154

most accurate k =   16
testing accuracy =   0.6979591836734694
```

based on the result we get. our validation accuracy is highest when $k = 16$
(This is just based on one single test result, for different runs, the output can varies.)

The testing accuracy ≈ 0.698.

1.c



J.C

The difference in accuracy is result in the difference between 'cosin similarity' and 'Euclidean difference'

Euclidean distance function is $\sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$

It calculate the physical distance between two vector points.

Cosin similarity function is $\frac{X \cdot Y}{|X||Y|}$

It calculate the angel between two vectors.

The reason why 'cosin' is more accurate the Euclidean is there are some trivial words occur too many times in our titles like: 'to' 'is', 'will' ... we can't determine true or fake by these words, but they influence the result.

To make it easy to understand, lets say we have two titles: 'fake news' and 'fake news $\underbrace{\text{news} \cdots \text{news}}_{100 \text{ times}}$', by vectorize them we have $\begin{bmatrix}1\\1\end{bmatrix}$ and $\begin{bmatrix}1\\100\end{bmatrix}$.

They are both fake news, but by Euclidean,

The distance between them is pretty large, when news occurs 1000 times it becomes even larger.

but by 'cosin', the difference between news occur 2 times, 100 times even billion times is pretty small. When our dimension is large, it becomes even less important comparing to 'Euclidean'

2a

$$\frac{\partial J^{\beta}_{reg}}{\partial w_j} = \frac{1}{N}\sum_{i=1}^{N} x_j^i (y^i - t^i) + \beta_j w_j$$

$$w_j \leftarrow w_j - \alpha \frac{\partial J^{\beta}_{reg}}{\partial w_j}$$

$$\underline{\underline{w_j \leftarrow w_j(1-\alpha\beta_j) - \frac{\alpha}{N}\sum_{i=1}^{N} x_j(y^i - t^i)}}$$

$$\frac{\partial J^{\beta}_{reg}}{\partial b} = \frac{1}{N}\sum_{i=1}^{N} 1 \cdot (y^i - t^i)$$

$$b \leftarrow b - \alpha \frac{\partial J^{\beta}_{reg}}{\partial \beta}$$

$$b \leftarrow b - \frac{\alpha}{N}\sum_{i=1}^{N} (y^i - t^i)$$

Because only the decay of weight is related to β, and the decay of b is same with or without regularization.

2b. $J_{reg}^{\beta} = \frac{1}{2N}\sum_{i=1}^{N}\left(\sum_{j=1}^{D} w_j x_j^{(i)} - t^{(i)}\right) + \frac{1}{2}\sum_{j=1}^{D}\beta_j w_j^2$

$$\frac{\partial J_{reg}^{\beta}}{\partial w_j} = \frac{1}{N}\sum_{i=1}^{N} x_j^{(i)}\left(\sum_{j'=1}^{D} w_{j'} x_{j'}^{(i)} - t^{(i)}\right) + \beta_j w_j$$

$$= \frac{1}{N}\sum_{j'=1}^{D}\left(\sum_{i=1}^{N} x_j^{(i)} x_{j'}^{(i)}\right) w_{j'} - \frac{1}{N}\sum_{i=1}^{N} x_j^{(i)} t^{(i)} + \beta_j w_j$$

$$= \sum_{j'=1}^{D}\left(\frac{1}{N}\sum_{i=1}^{N} x_j^{(i)} x_{j'}^{(i)} + \frac{\beta_j}{D}\right) w_{j'} - \frac{1}{N}\sum_{i=1}^{N} x_j^{(i)} t^{(i)}$$

We have

$$A_{jj'} = \frac{1}{N}\sum_{i=1}^{N} x_j^{(i)} x_{j'}^{(i)} + \beta_j$$

$$c_j = \frac{1}{N}\sum_{i=1}^{N} x_j^{(i)} t^{(i)}$$

2c. $A = \sum_{j'=1}^{D} A_{jj'}$

$= \frac{1}{N} X^T X + \beta$ where $\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_D \end{bmatrix}$

$C = \frac{1}{N} X^T t$

$AW = C$

$\frac{1}{N} X^T X W + \beta W = \frac{1}{N} X^T t$

$X^T X W + N\beta W = X^T t$

$W(X^T X + N \mathrm{diag}(\beta)) = X^T t$

$W = (X^T X + N \mathrm{diag}(\beta))^{-1} X^T t$

3. $$y = Xw + b\cdot 1$$

$$\frac{\partial J}{\partial y} = \frac{1}{N}\sum_{i=1}^{N} \sin(y-t)$$

since We have

$$y = Xw + b\cdot 1$$

$$y = \begin{bmatrix} x_1^{(1)} & \cdots & x_D^{(1)} \\ x_1^{(2)} & & x_D^{(2)} \\ \vdots & \ddots & \vdots \\ x_1^{(N)} & \cdots & x_D^{(N)} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_D \end{bmatrix} + \begin{bmatrix} b \\ b \\ \vdots \\ b \end{bmatrix}$$

$$\frac{\partial y}{\partial w} = \begin{bmatrix} \frac{\partial y_1}{\partial w_1} & \frac{\partial y_2}{\partial w_1} & \cdots & \frac{\partial y_D}{\partial w_1} \\ \frac{\partial y_1}{\partial w_2} & & \ddots & \frac{\partial y_D}{\partial w_2} \\ \vdots & & & \vdots \\ \frac{\partial y_1}{w_D} & \cdots & & \frac{\partial y_D}{\partial w_D} \end{bmatrix}$$

$$= \begin{bmatrix} X_1^{(1)} & \cdots & X_1^{(N)} \\ X_2^{(1)} & \ddots & \vdots \\ \vdots & & \vdots \\ X_D^{(1)} & \cdots & X_D^{(N)} \end{bmatrix} = X^T$$

$$\frac{\partial J}{\partial W} = \frac{\partial J}{\partial y} \cdot \frac{\partial y}{\partial W}$$

$$= X^T \cdot \left(\frac{1}{N} \sum_{i=1}^{N} \sin(XW + b \cdot 1 - t) \right)$$

Same as above

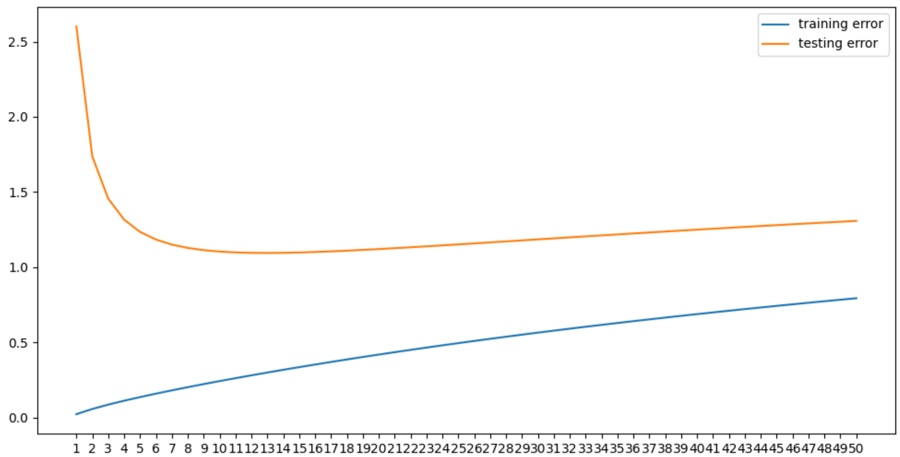
We have

$$\frac{\partial y}{\partial b} = 1^T$$

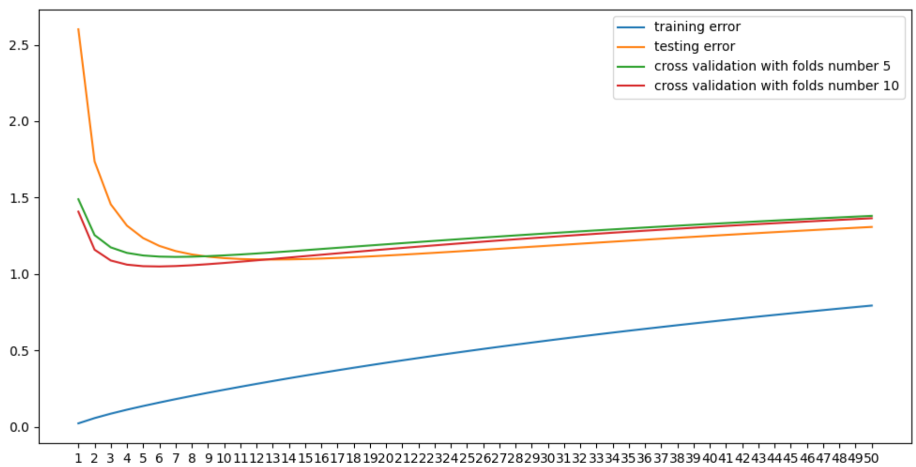
$$\frac{\partial J}{\partial b} = \frac{\partial J}{\partial y} \cdot \frac{\partial y}{\partial b}$$

$$= 1^T \left(\frac{1}{N} \sum_{i=1}^{N} \sin(XW + b \cdot 1 - t) \right)$$

4 c.



d



based on the result.
$\lambda = 10$ has lower error rate.