`

Question 1

The likelihood of observed document is

$$P(d_1,\ldots,d_N) = \prod_{n=1}^{N}\sum_{k=1}^{K} p(z_{n,k}=1, d_n)$$

$$= \prod_{n=1}^{N}\sum_{k=1}^{k}(\varphi_k \prod_{w\in A} \mu_{k,w}^{c(w,d_n)})$$

Where φ_k is the prior probability of the cluster

$u_{k,w}$ is the proportion of word w in cluster k

$c(w,d_n)$ is the count of word w in document d

The log likelihood is therefore

$$\sum_{n=1}^{N} ln \sum_{k=1}^{k}(\varphi_k \prod_{w\in A} u_{k,w}^{c(w,d_n)})$$

To maximize the log likelihood, we can maximize the Q function which is a lower bound of the log likelihood

$$\begin{aligned} Q(\boldsymbol{\theta},\boldsymbol{\theta}^{\text{old}}) &:= \sum_{n=1}^{N}\sum_{k=1}^{K} p(z_{n,k}=1|d_n,\boldsymbol{\theta}^{\text{old}})\ln p(z_{n,k}=1,d_n|\boldsymbol{\theta}) \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K} p(z_{n,k}=1|d_n,\boldsymbol{\theta}^{\text{old}})\left(\ln\varphi_k + \sum_{w\in\mathcal{A}} c(w,d_n)\ln\mu_{k,w}\right) \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(z_{n,k})\left(\ln\varphi_k + \sum_{w\in\mathcal{A}} c(w,d_n)\ln\mu_{k,w}\right) \end{aligned}$$

$\boldsymbol{\theta} := (\boldsymbol{\varphi},\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_K)$ is the collection of model parameters

$\gamma(z_n,k) := p(z_{n,k}=1|d_n,\boldsymbol{\theta}^{\text{old}})$ is the conditional probability (or posterior probability) of cluster k given a document d, where $\sum_{k=1}^{K}\gamma(z_{nk})=1$.

The posterior probability $\gamma(z_n,k)$ is computed as follows:

$$\gamma(z_n,k) = \frac{\varphi_k \prod_{w\in A} u_{k,w}^{c(w,d_n)}}{\sum_{k=1}^{k}(\varphi_k \prod_{w\in A} u_{k,w}^{c(w,d_n)})}$$

However, we should be aware that $\prod_{w\in A} u_{k,w}^{c(w,d_n)}$ can easily result in arithmetic

`

underflow therefore we need to do some scaling.

We define z_k = ln $(\varphi_k \prod_{w\in A} u_{k,w}^{c(w,d_n)})$ = ln(φ_k) + $\sum c(w,d_n)\ln(\mu_{k,w})$

So $\gamma(z_n, k)$ = $\frac{e^{z_k}}{\sum_{k=1}^{K} e^{z_k}}$

Now let m = max(z_k)

Multiplying the numerator and denominator of $\gamma(z_n, k)$ by e^{-m} gives

$$\frac{e^{z_k - m}}{\sum_{k=1}^{K} e^{z_k - m}}$$

To maximize the Q function, we can form the Lagrangian to enforce the constraints where $\sum_{k=1}^{K} \varphi_k = 1$ and $\sum_{w\in\mathcal{A}} \mu_{k,w} = 1$ and set the derivatives to zero which leads to the following solution for the model parameters:

$\varphi_k = \frac{N_k}{N}$ where $N_k := \sum_{n=1}^{N} \gamma(z_{n,k})$

$$\mu_{k,w} = \frac{\sum_{n=1}^{N} \gamma(z_{n,k}) c(w, d_n)}{\sum_{w'\in\mathcal{A}} \sum_{n=1}^{N} \gamma(z_{n,k}) c(w', d_n)}$$

We now have all the parameters that are required to implement the hard EM algorithm

Parameters initialization:

Since φ_k and $\mu_{k,w}$ is unknown, we will initialize these 2 parameters uniformly.

E step:

- Compute $\gamma(z_n, k)$ base on the old φ_k and $\mu_{k,w}$ for each document
- Find the posterior probability that has the largest value and set it equals to 1 and all others to 0.

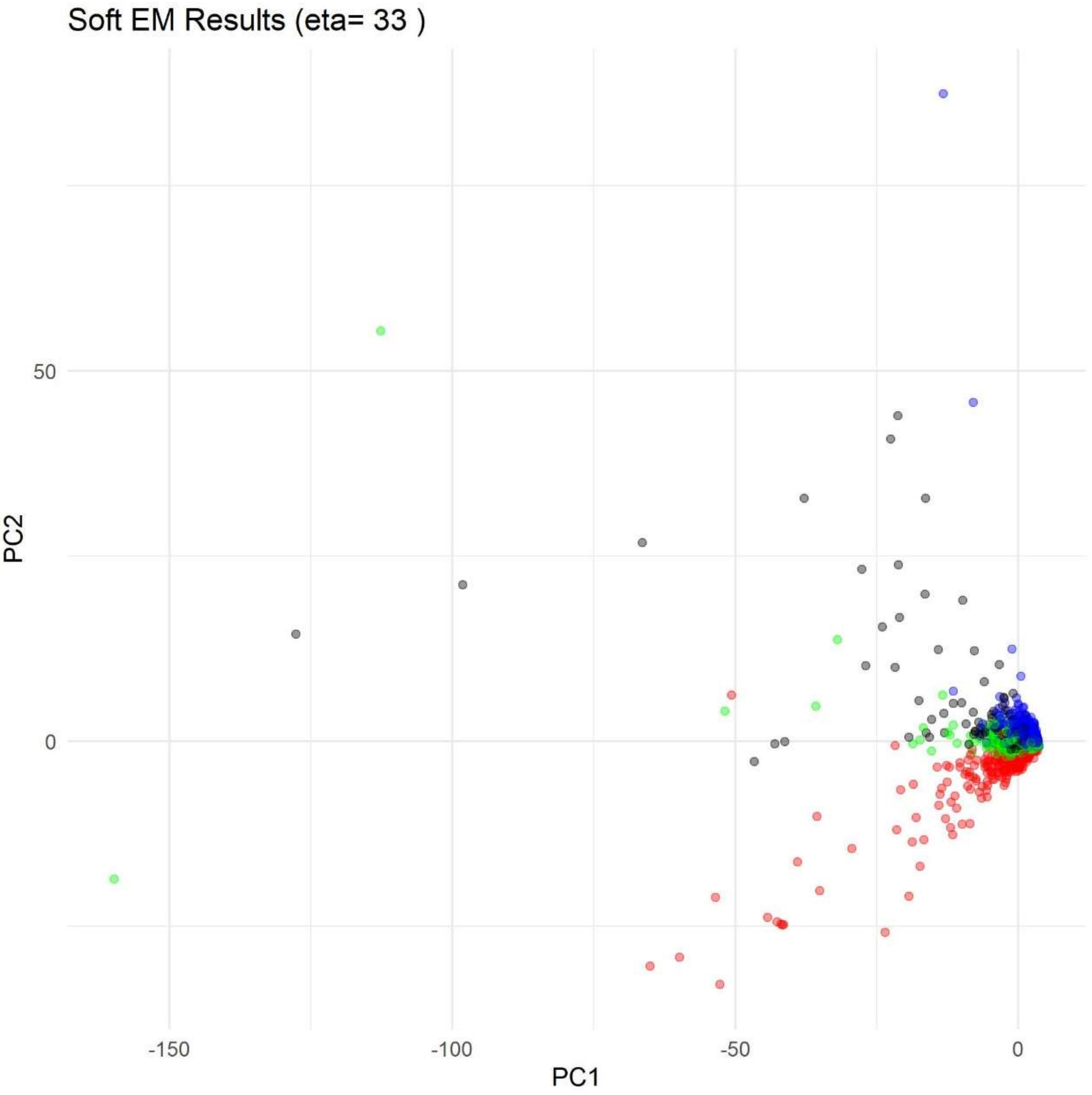
M step:

Once we have computed the gamma, we can substitute gamma into the above formula and update φ_k and $\mu_{k,w}$ respectively.

`

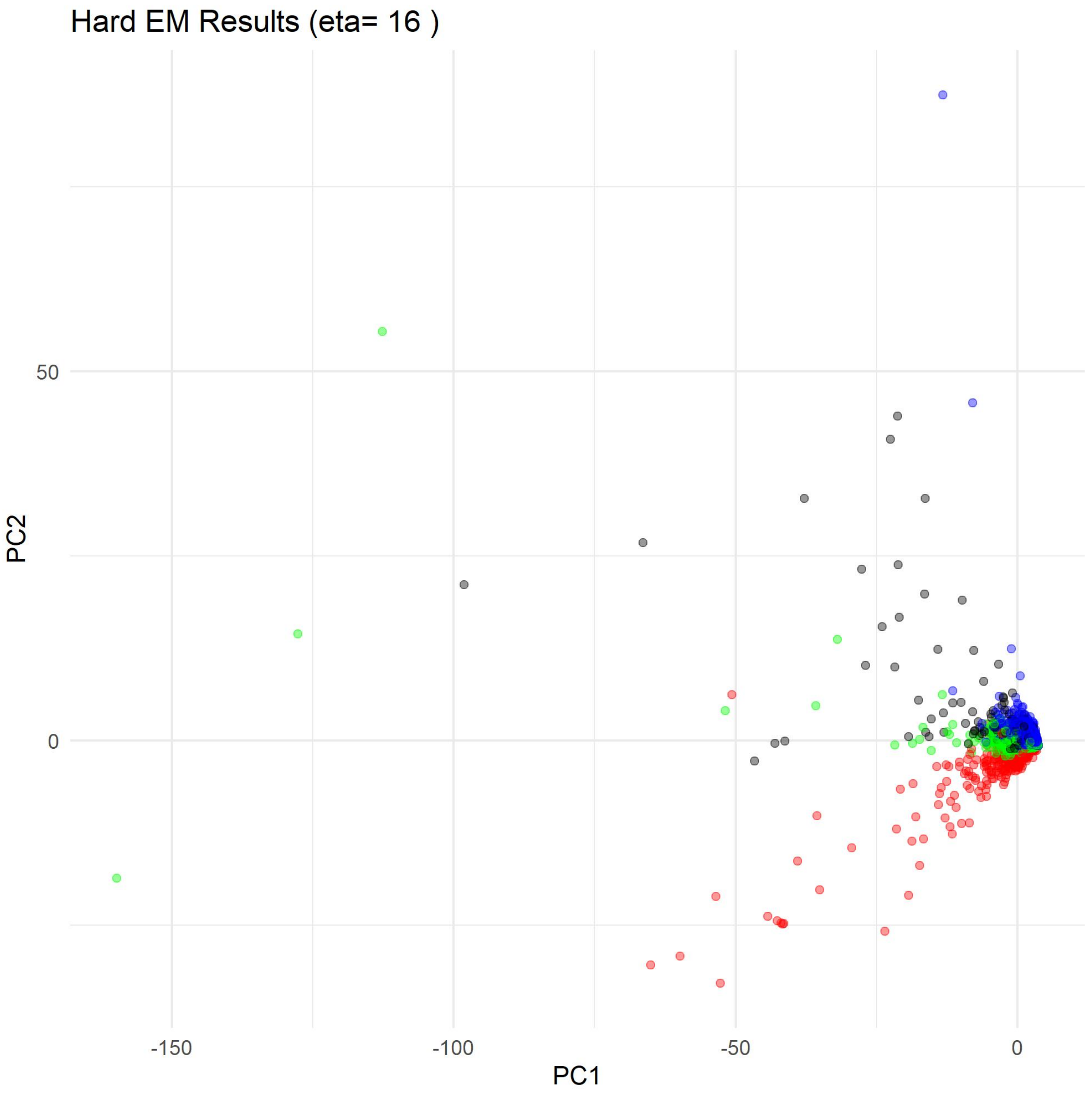
Both of the E and M steps keep repeating until convergence is achieved.

Visualization of soft EM



`

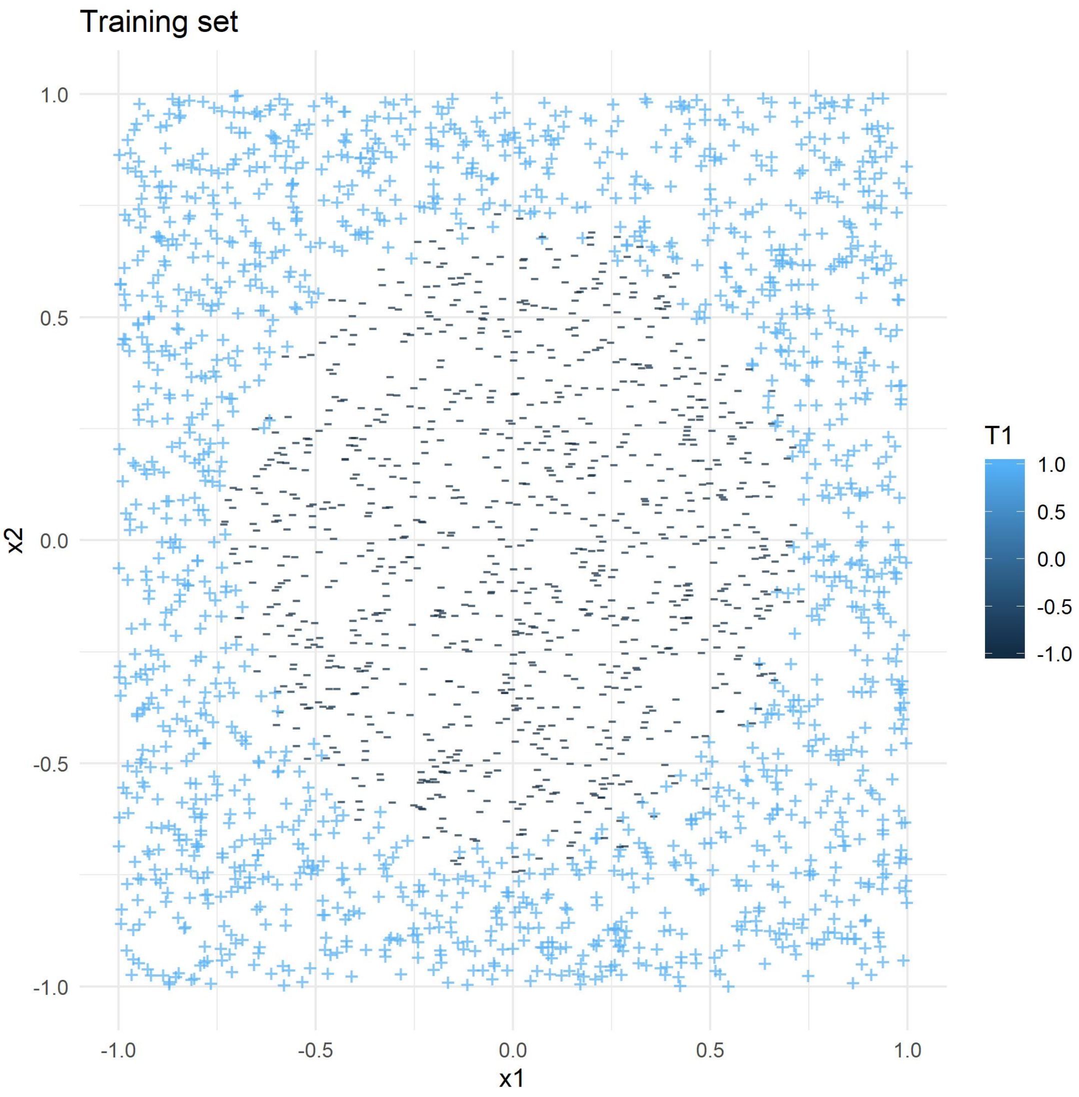
Visualization of hard EM



`

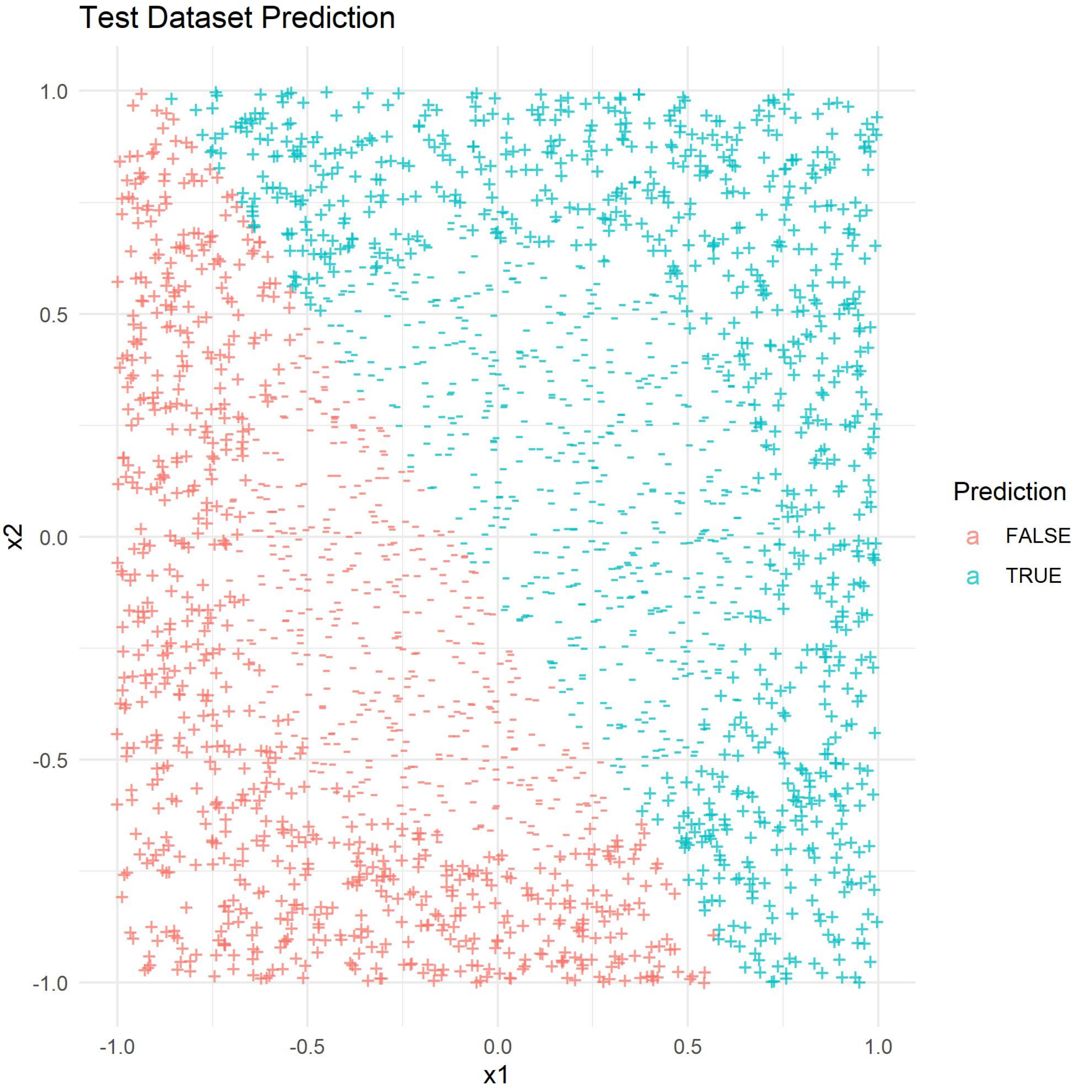
Question 2

I. Plot training data



`

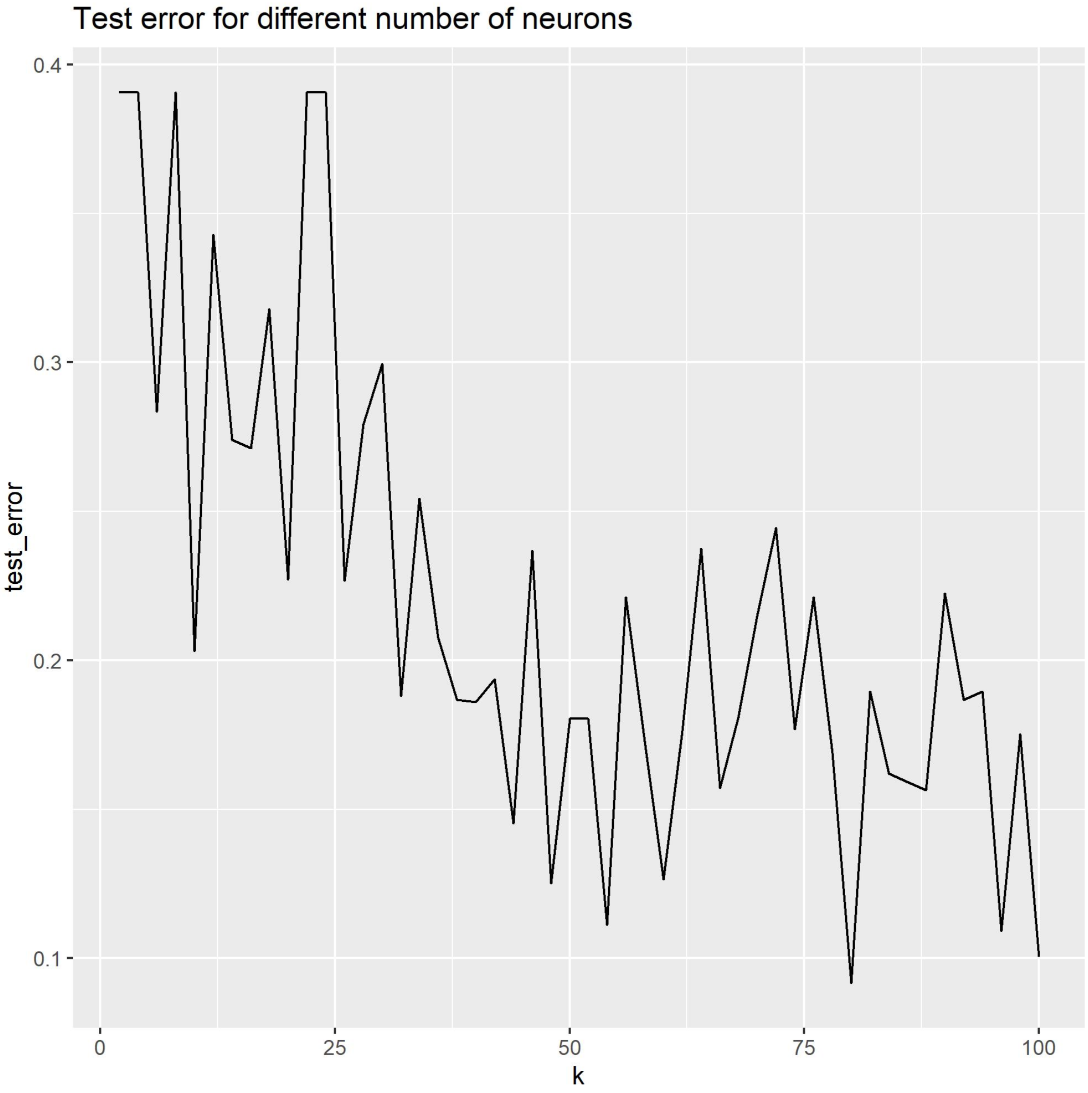
II. Perceptron



The test error for the perceptron classifier is 51.4%.

`

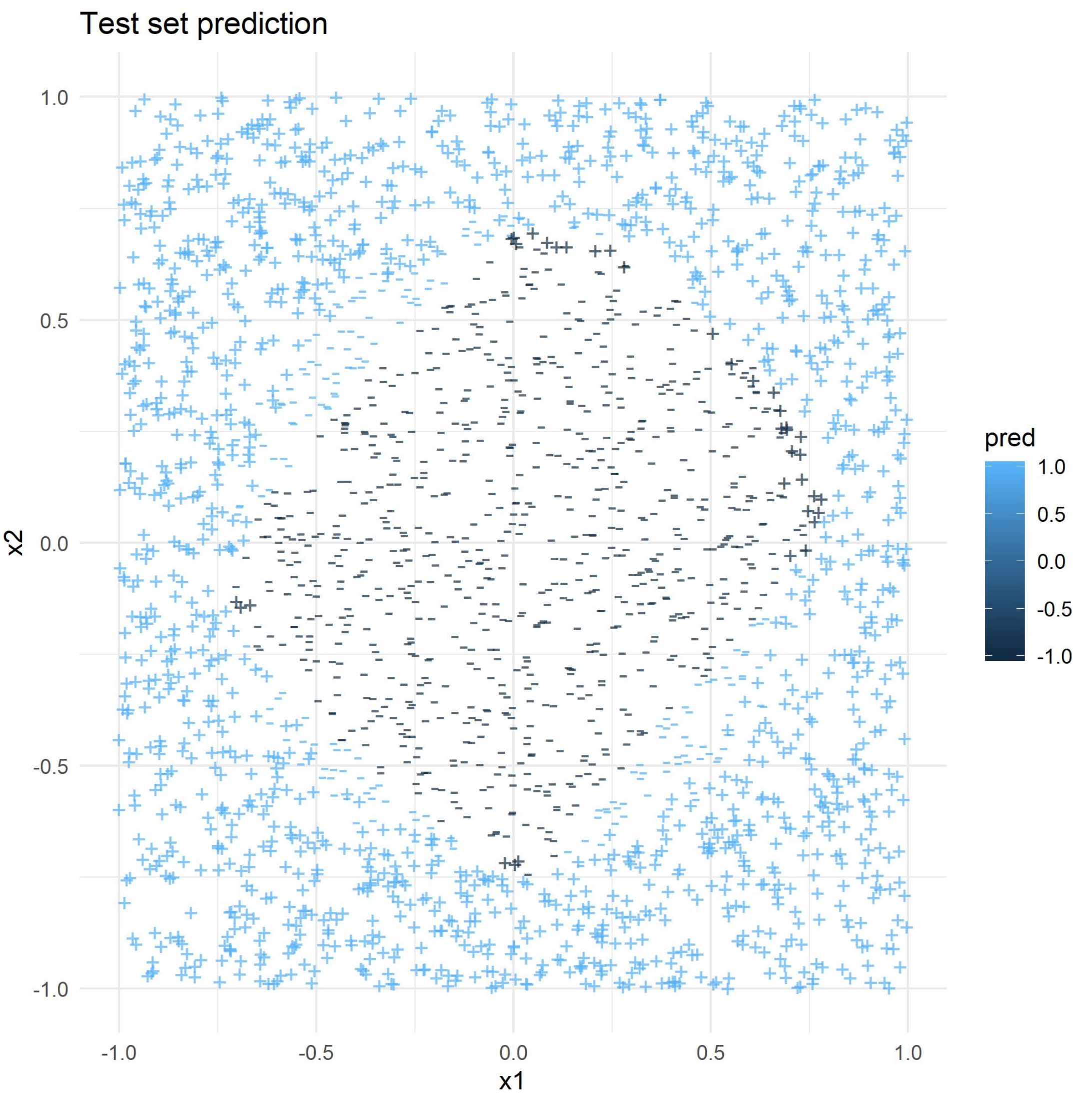
III. Neural network



The model with 80 neurons has the lowest test error.

`

Prediction using model with 80 neurons



IV. Test error of perceptron and all variants of neural network

Neural network	
K	Test error %
2	39.08
4	39.08
6	28.36
8	39.08
10	20.32
12	34.28
14	27.4
16	27.12
18	31.8
20	22.72

`

22	39.08
24	39.08
26	22.68
28	27.92
30	29.96
32	18.8
34	25.44
36	20.76
38	18.68
40	18.6
42	19.36
44	14.52
46	23.68
48	12.52
50	18.04
52	18.04
54	11.12
56	22.12
58	17.28
60	12.64
62	17.64
64	23.76
66	15.72
68	18.08
70	21.48
72	24.44
74	17.68
76	22.12
78	16.88
80	**9.16**
82	18.96
84	16.2
86	15.92
88	15.64
90	22.24
92	18.68
94	18.96

96	10.92
98	17.52
100	10.04
Perceptron	
-	51.4

V. Reasons for the difference between perceptron and neural network

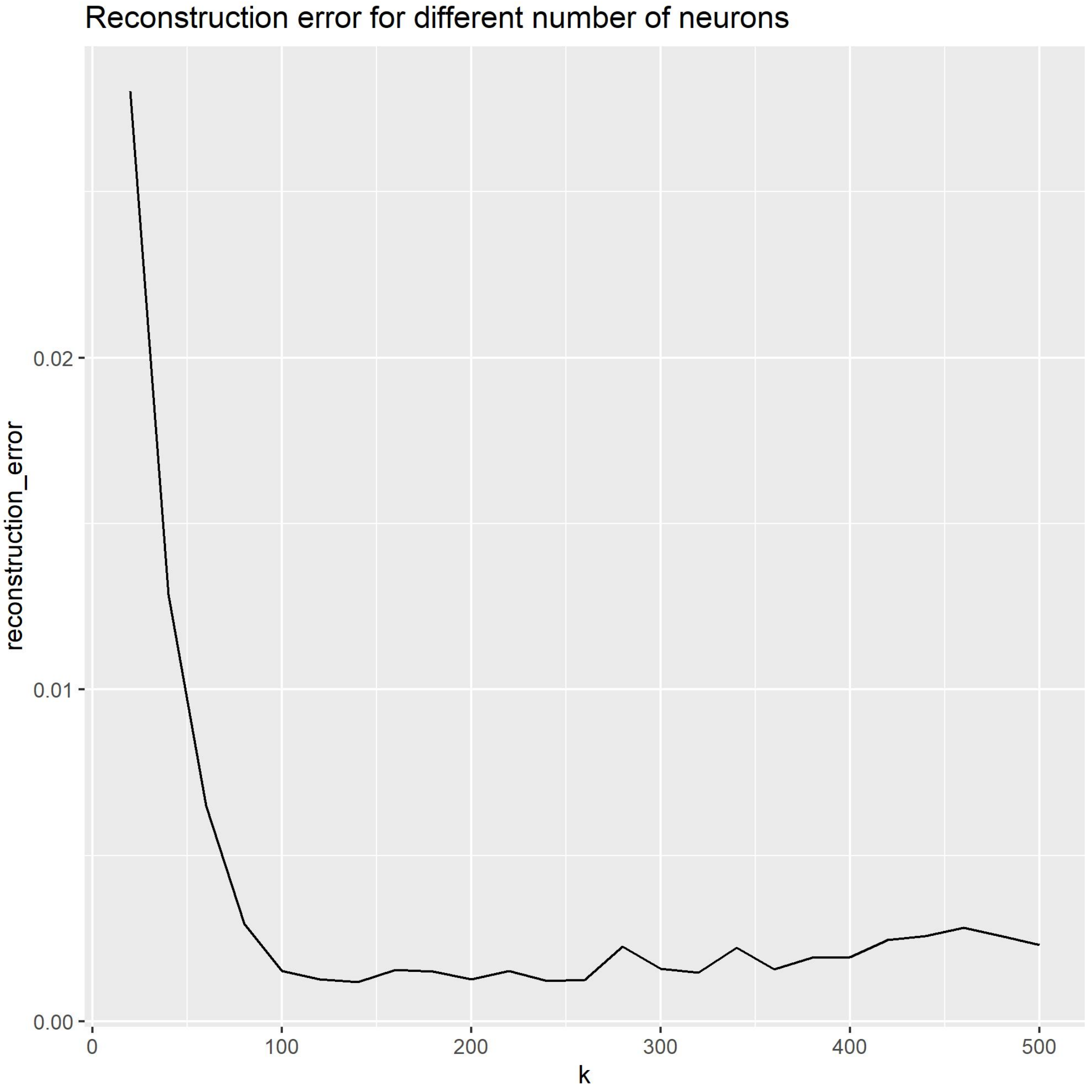
Both the training data and test data are not linearly separable. Since the perceptron is a linear classifier, its decision boundary is always a straight line. No matter how we place a straight line on the data, roughly half of the data will be misclassified, this is why the test error of perceptron classifier is 51.4%.

All the variants of the 3 layer neural network perform better than perceptron. Since neural network can approximate almost any function, it can estimate the target decision boundary to any required precision. Therefore, it can be both a linear or non-linear classifier and it outperforms perceptron in this case.

If the data is linearly separable, the perceptron will be as good as the neural network.

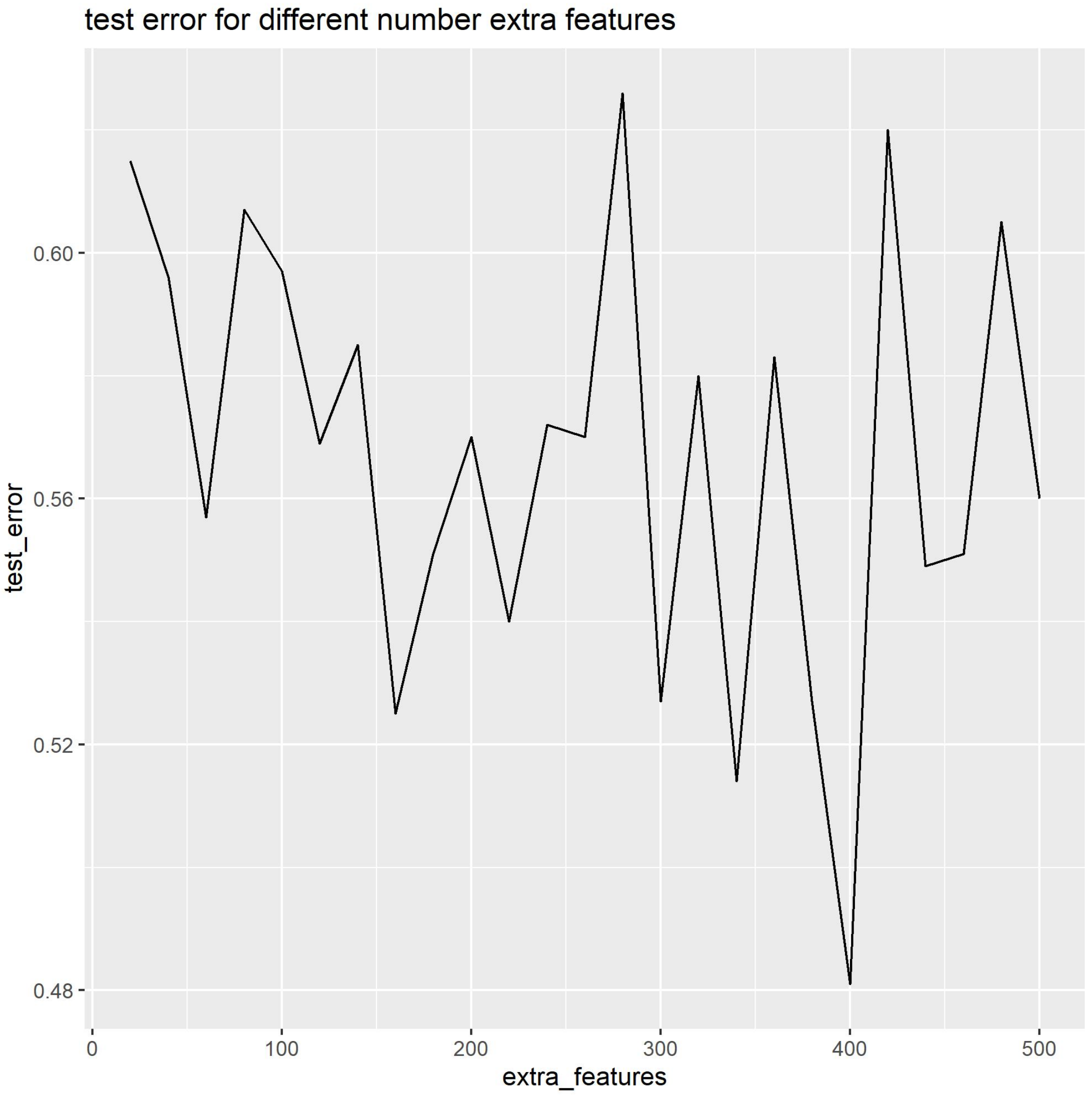
`

Question 3



The reconstruction error is high for low value of k, this is because the number of useful features is larger than k in this range. By mapping the useful features to a lower dimension, the usefulness of the features is lost. The reconstruction error for K=140 to K = 500 is roughly the same, this is because we have enough neurons in the hidden layer to map the important features.

`



The optimal number of neurons for reconstruction error is k=140 while the optimal number of neurons for reconstruction error is k=400.

Generally speaking, both reconstruction error and test error drop when k increases. For test error, this is because the extra features are extracted from the autoencoder which can help the neural network to learn better. However, having too much extra features hurt the model performance because of overfitting. It is worth mentioning that, for low number of extra features, the performance is actually worse than no extra feature, this is because the autoencoder compress the useful features too much and the information of the useful features is lost.

For reconstruction error, it is expected that it will decrease when k increases, and reconstruction error can be 0 if the hidden layer is of the same size of the input layer.

`

The fluctuation of reconstruction error is due to random initialization of model parameters.