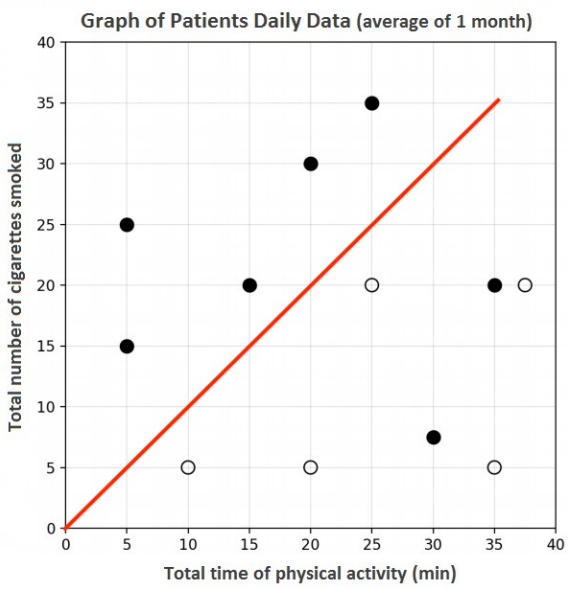
question 1)
a)



b) first most nearest point to (5,21) is (5,20) which has lung cancer. then this point is classified as with lung cancer.

c) first of all we transform the data to new space:

x	y	label	new space
5	15	1	0.7 * (20)
5	25	1	0.7 * (30)
15	20	1	0.7 * 35
20	30	1	0.7 * 50
25	35	1	0.7 * 60
30	7.5	1	0.7 * 37.5
35	20	1	0.7 * 55
10	5	0	0.7 * 15
20	5	0	0.7 * 25
25	20	0	0.7 * 45

x	y	label	new space
35	5	0	0.7 * 40
37.5	20	0	0.7 * 57.5
5	21	0	0.7 * 26

label of the most nearest point to the the given point is healthy. Then this point is classified as healthy.

d) No. because in PCA data is transformed to the new space without considering label of the points and projects them to the direction of maximum variance. That means it doesn't project data to a new space where different classes are separated well from each other.

e, f)

	gene 1	gene 2	gene 3
mean	1.3	6	7.6
var	9.21	0.8	0.840

var is calculated as follows:

var(x1) = 1/10 * (1 + 36 + 1 + 4 + 0 + 1 +1 + 4 + 25 + 36) - 1.3*1.3
var(x2) = 1/10 * (36 + 36 + 25 + 36 + 64 + 25 + 36 +25 + 36 +49) - 36
var(x3) = 1/10 * (49 + 81 + 49 + 49 + 64 + 81 + 49 + 64 + 64 + 36) - 7.6 * 7.6

g and h)

g) $\bar{X}$ is:

-۲,۳	۴,۷	-۲,۳	۰,۷	-۱,۳	-۳,۳	-۲,۳	-۳,۳	۳,۷	۴,۷
۰	۰	-۱	۰	۲	-۱	۰	-۱	۰	۱
-۰,۶	۱,۴	-۰,۶	-۰,۶	۰,۴	۱,۴	-۰,۶	۰,۴	۰,۴	-۱,۶

h) C =

۹,۲۱	۱	-۰,۵۸
۱	۰,۸	-۰,۲
-۰,۵۸	-۰,۲	۰,۸۴

we mus calculate $\bar{X}\bar{X}^T \times \frac{1}{n}$ for filling this matrix

C_{12} = (-۲,۳ × -۱ + ۲ × -۱,۳ - ۱ × -۲,۳ - ۱ × -۳,۳ + ۱ × ۴,۷) × ۱/۱۰

= (۲,۳ - ۲,۶ + ۲,۳ + ۳,۳ + ۴,۷) × ۱/۱۰ = ۱

C_{13} = ۱/۱۰ × (-۲,۳ × -۰,۶ + ۴,۷ × ۱,۴ - ۲,۳ × -۰,۶ + ۰,۷ × -۰,۶

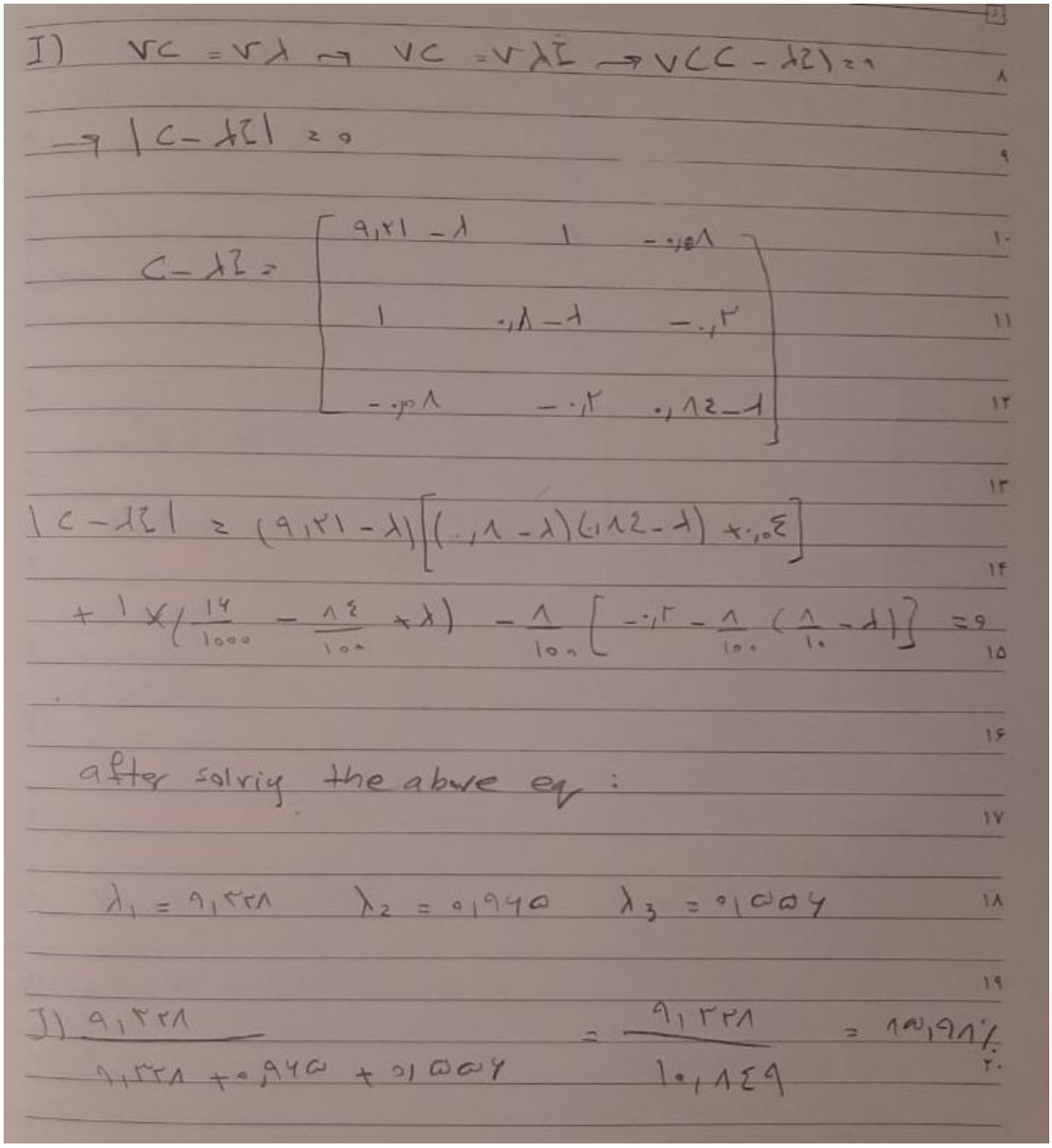
-۱,۳ × ۰,۴ - ۳,۳ × ۱,۴ - ۲,۳ × -۰,۶ - ۳,۳ × ۰,۴ + ۳,۷ × ۰,۴ + ۴,۷ × -۱,۶)

= -۰,۵۸

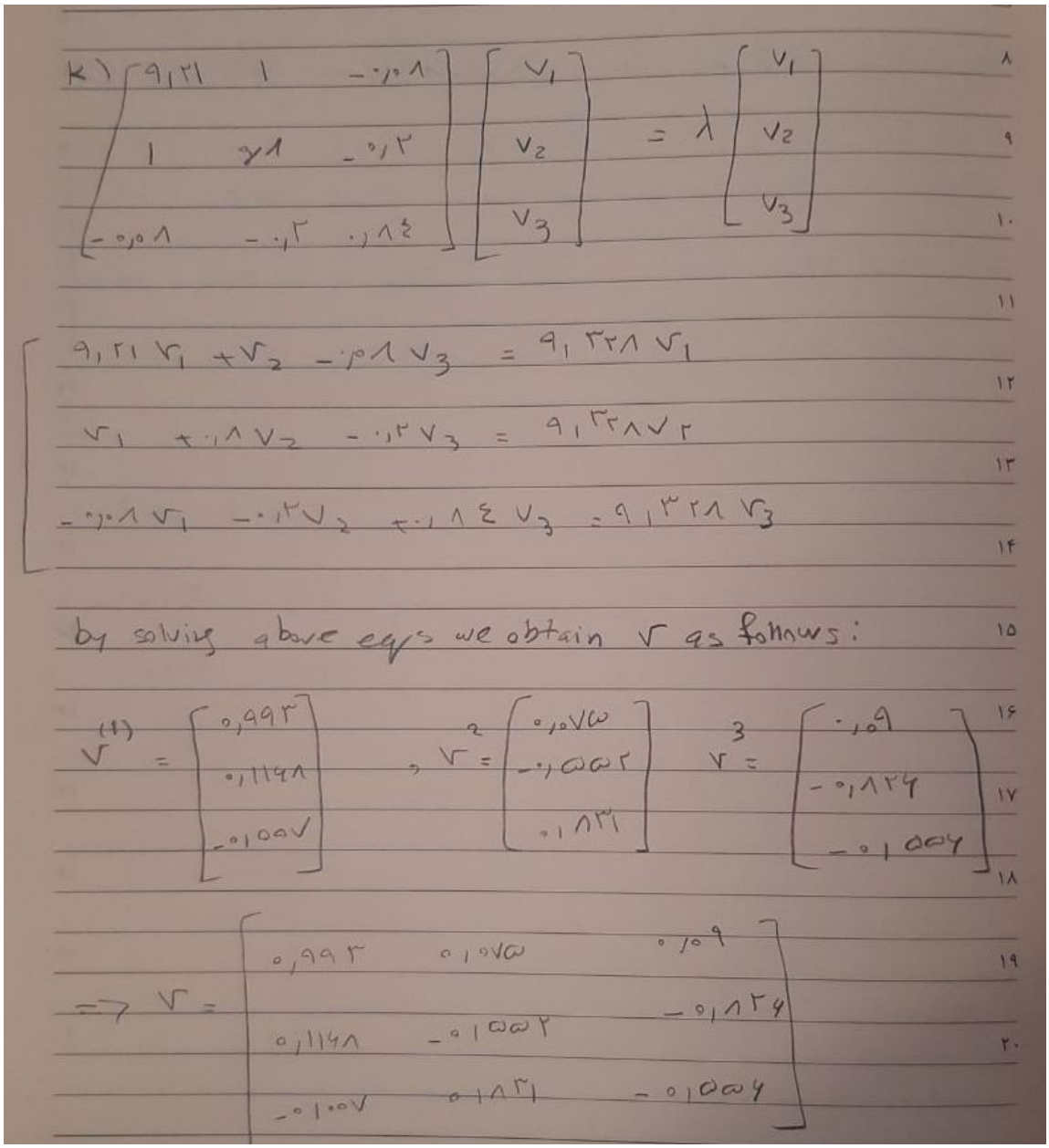
C_{23} = ۱/۱۰ (۰,۶ + ۰,۸ - ۱,۴ - ۰,۴ - ۱,۶) = -۰,۲

C_{11}, C_{22}, C_{33} are calculated in the previous part

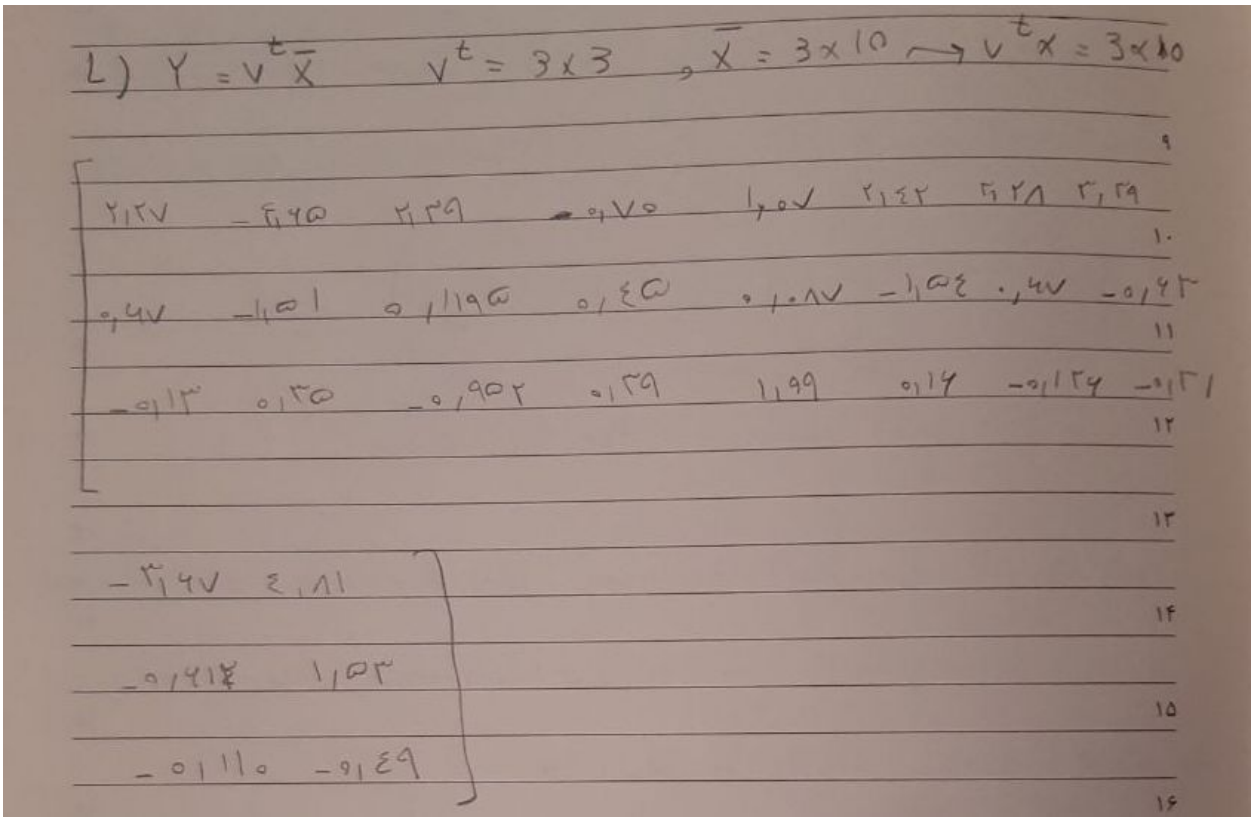
I and J)



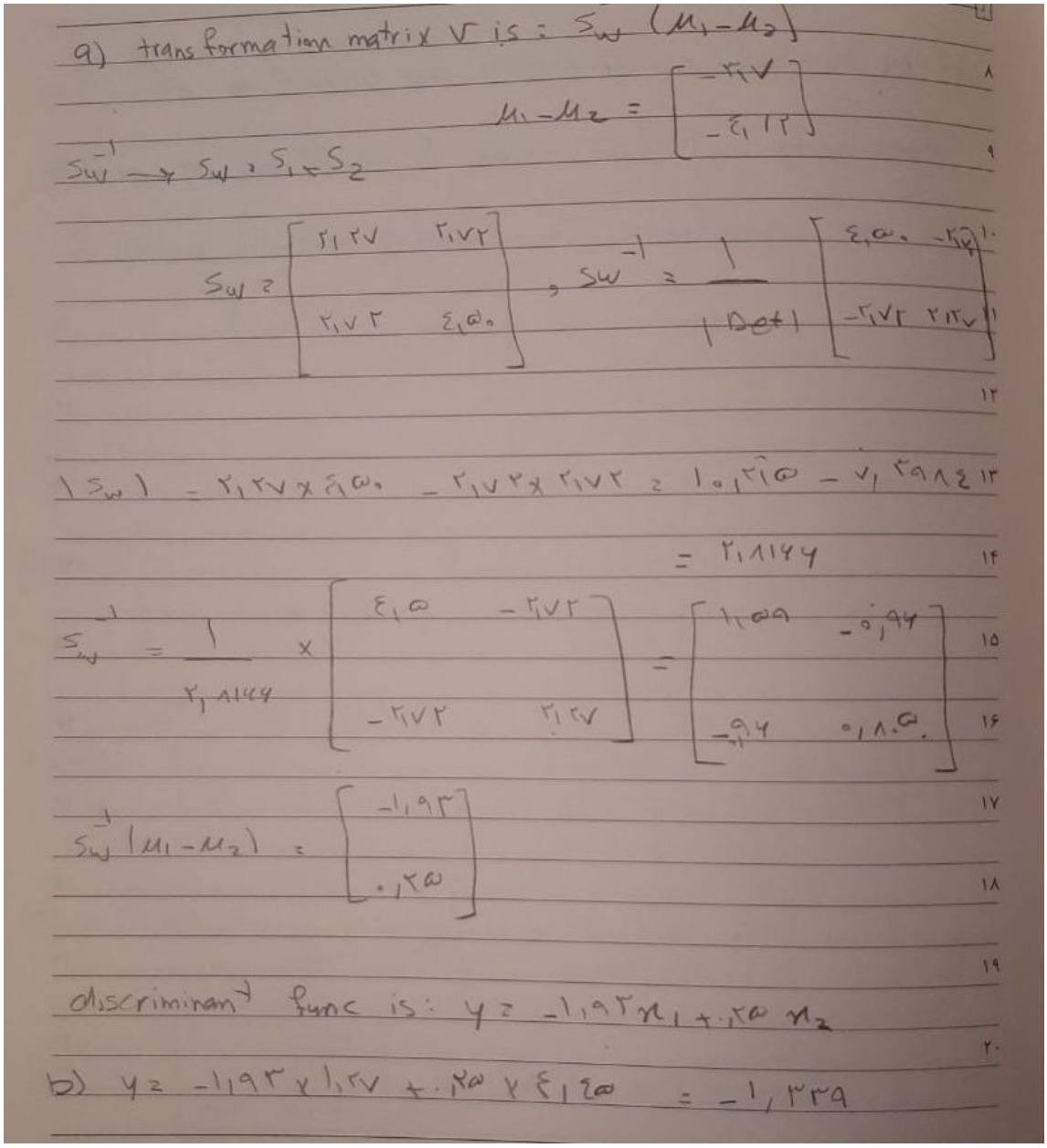
K)



L)



question 2)
A and B and C)



center of first class: $\mu_1^{new} = -1.9 \times 1.1 + 0.3 \times 3.19 = -3.08$

center of second class: $\mu_2^{new} = -1.9 \times 0.8 + 0.3 \times 1.01 = -1.819$

$|-1.339 + 3.58| < |-1.339 + 1.41|$

(-3.58 is μ_1^{new}, 1.41 is μ_2^{new})

→ this point belongs to the first class.

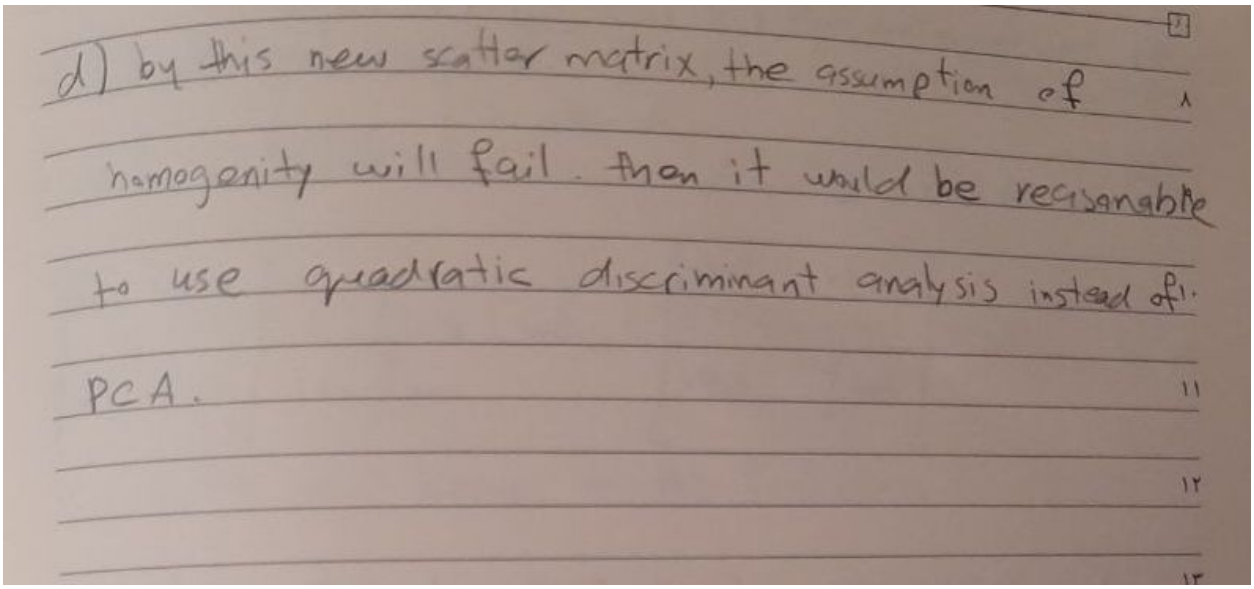
c) 1) independant variables are normally distributed

2) homogenity of variance/covariance: variances are the same across levels of predictors.

3) independance: participants are randomly sampled.

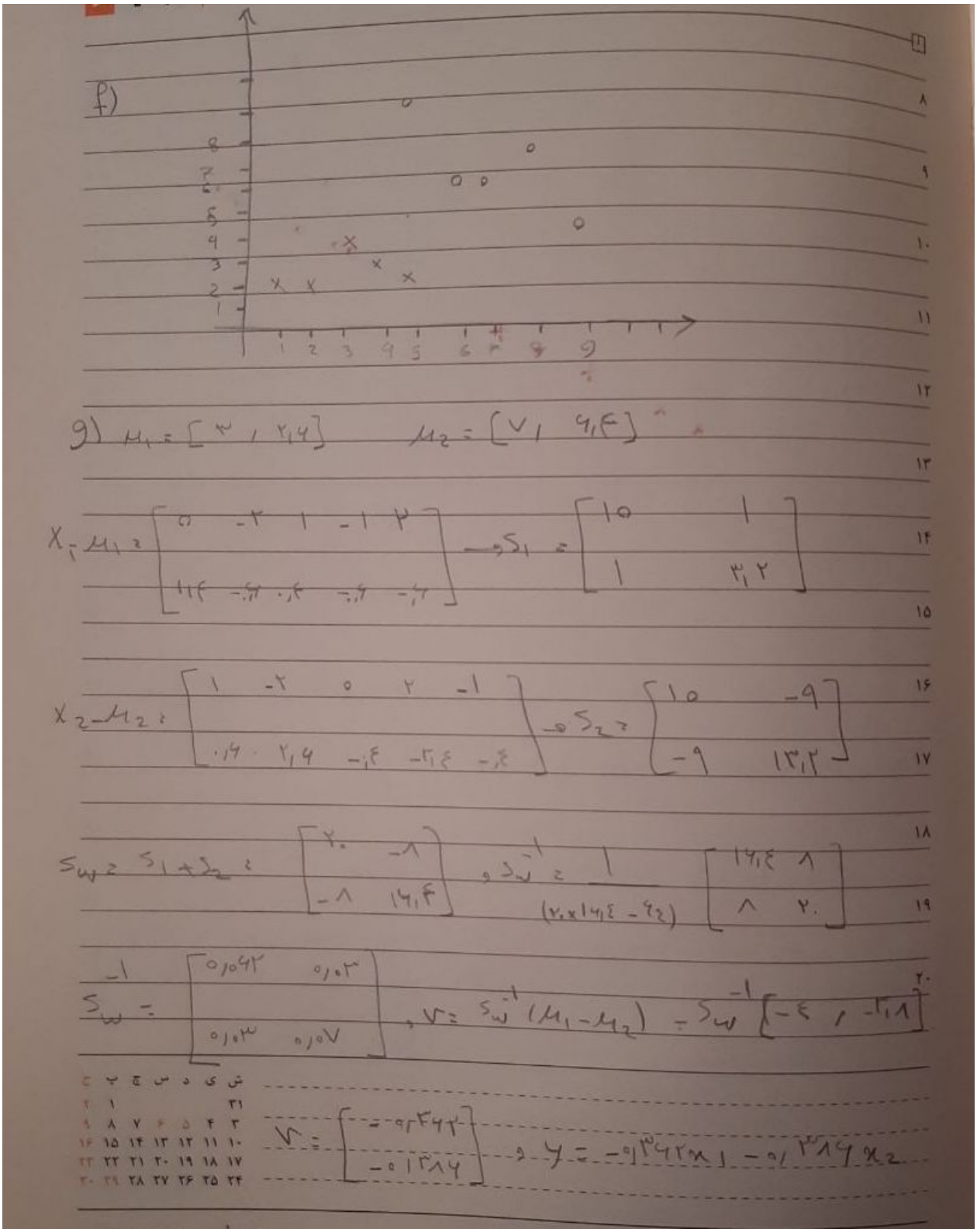
assumptions 1 and 3 are not clearly declared in the question. but assumption 2 is valid according to the given scatters.

D)

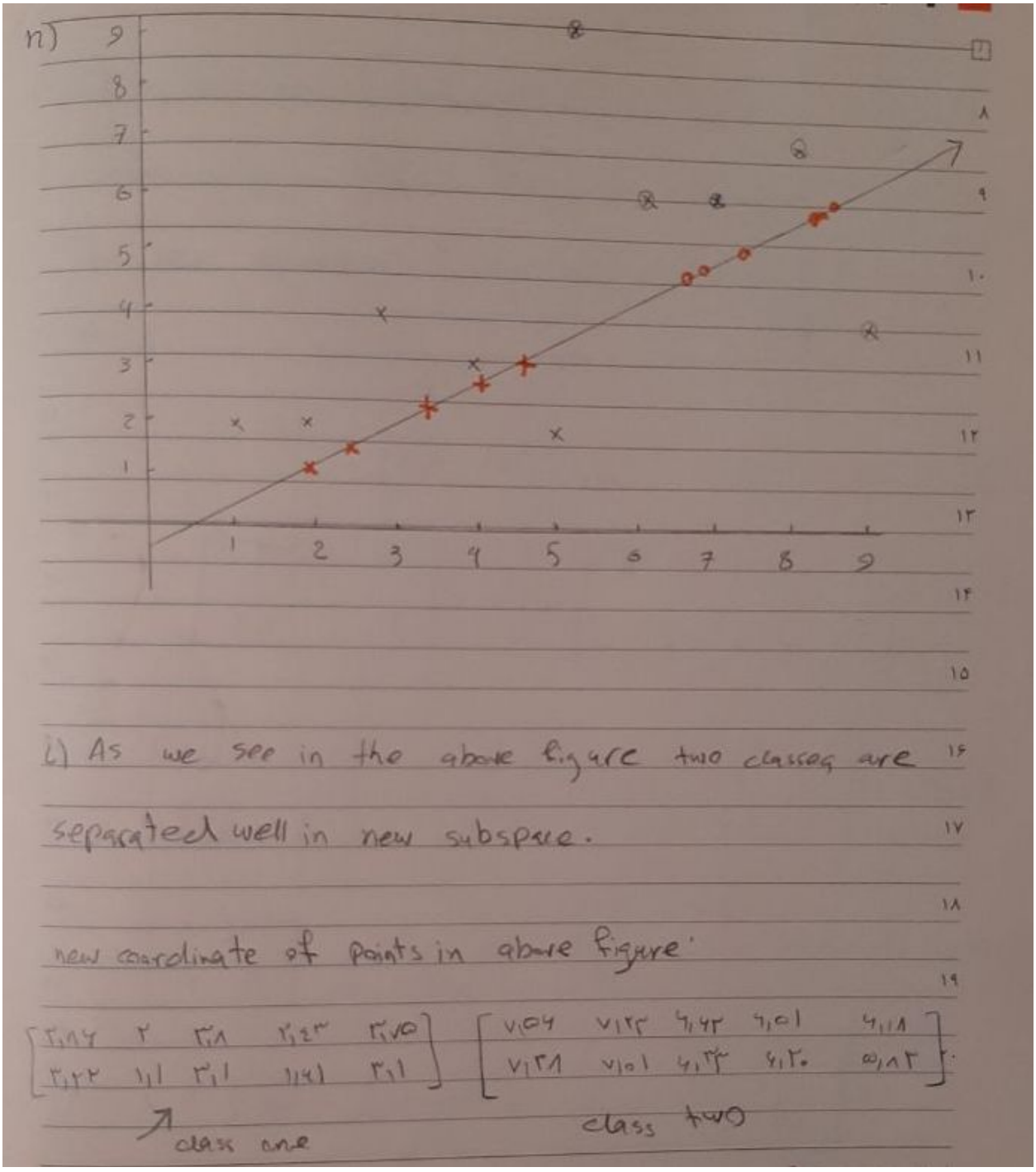

d) by this new scatter matrix, the assumption of
homogenity will fail. than it would be reasonable
to use quadratic discriminant analysis instead of
PCA.

e)

F and G)

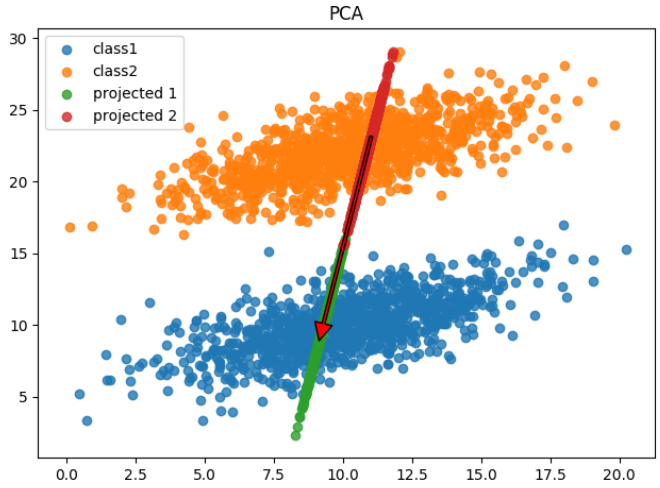


H and I)


n)
1 2 3 4 5 6 7 8 9
i) As we see in the above figure two classes are separated well in new subspace.
new coordinate of points in above figure:
class one
class two

question 3)

a, b, c)



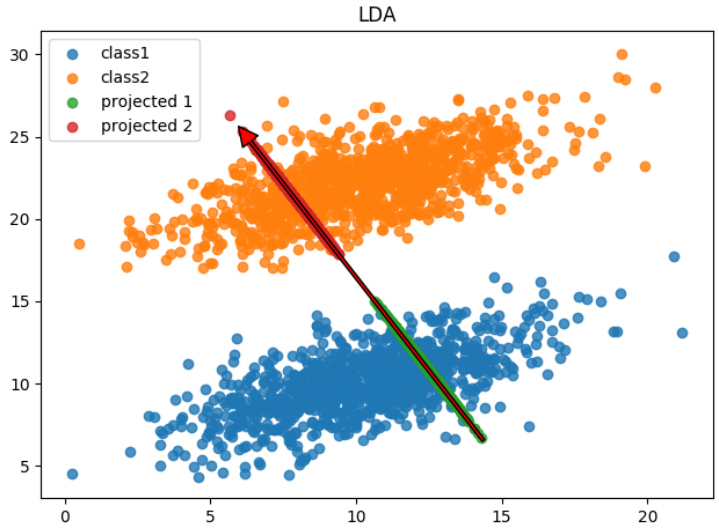
d) We expect that PCA projects the data to the line in direction of maximum variance, and as we see in the above plot, PCA has founded a projection line which is in the direction of maximum variance.

e) reconstruction error is obtained by following formula (MSE):

```
np.sqrt(np.sum([[x**2+y**2] for x,y in res]))
```

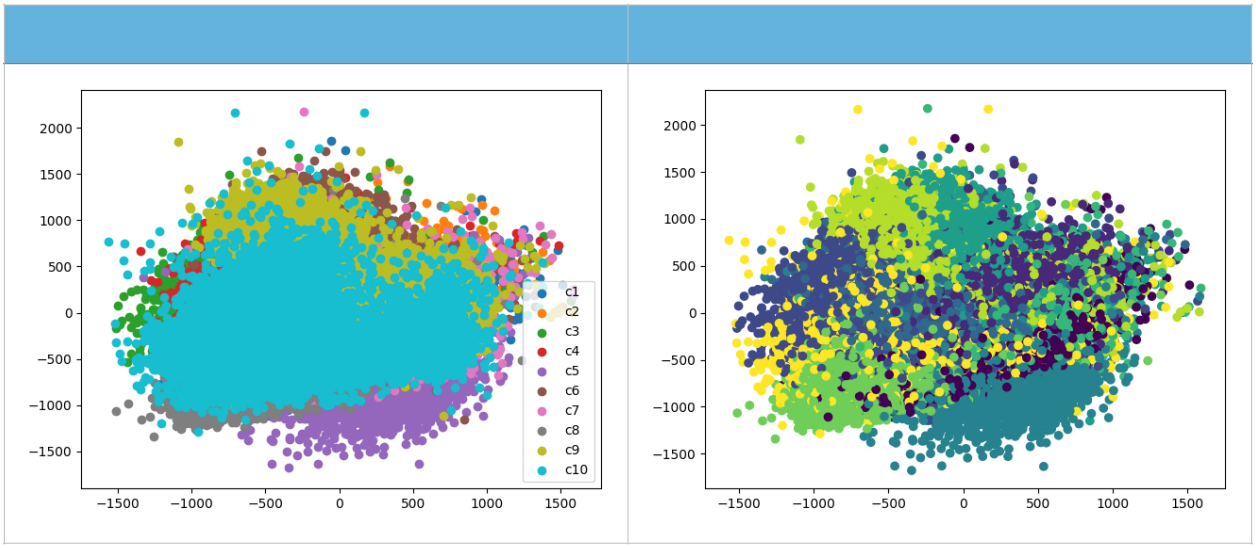
and is equal to: 17.2796

f, g)

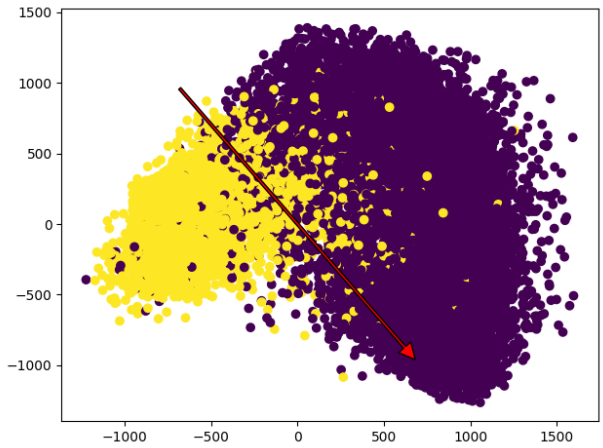


h) we expect that LDA projects data to the line which makes the most distinction between the two classes. As we see in the above figure, LDA has founded the projection line in the direction which most distinguishes the two classes. Actually it finds the line by maximizing the between class scatter and minimizing the within class scatter.

I) in the following plot, different classes are shown in the 2d space. As we see 10 different classes are not separated well and they are overlapping.



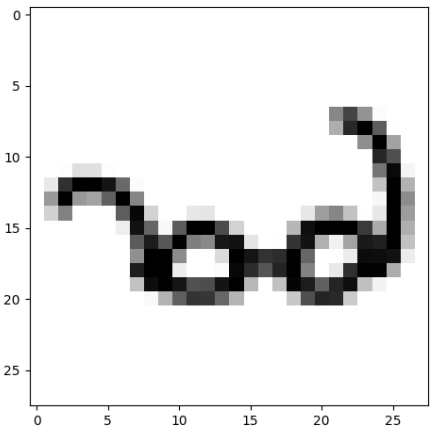
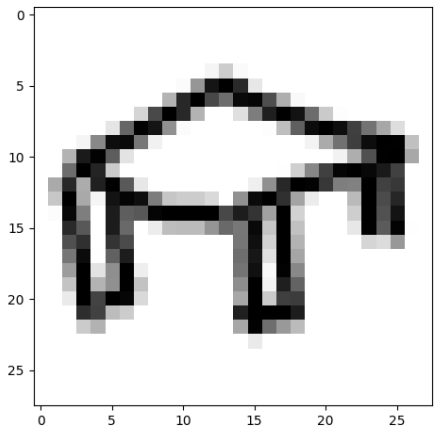
j, k) separator line is shown in the following figure:



and here are confusion matrixes:

train set	test set
[[7513 502] [315 7723]]	[[1853 132] [98 1864]]
0.94910	0.94172

l) most misclassified eyeglasses and pants are shown in the following table:

most misclassified eyeglass	most misclassified pant
	

m) required number of principal components for 90% variance is 170.

results of previous part with this number of principal components is below. As we see the performance of model is improved by using more principal components.

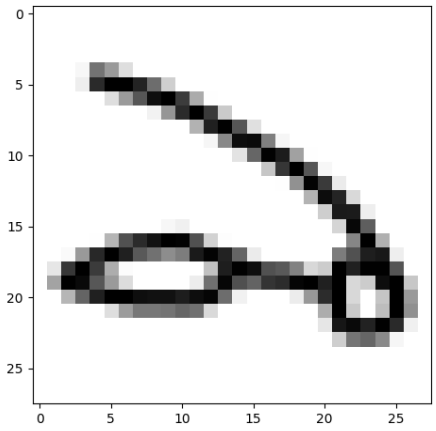
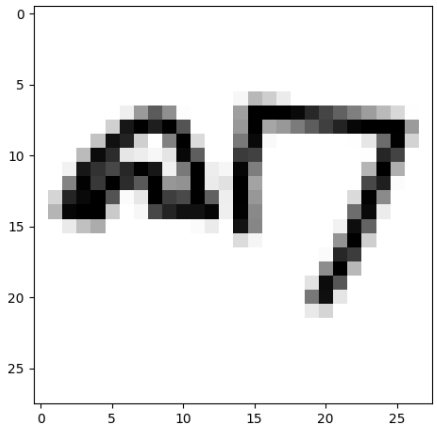
train set	test set
[[7791 224] [278 7760]]	[[1924 61] [76 1886]]
0.96872	0.9652900

N)

	1	3	5	7	9
train	[[7622 393] [405 7633]]	[[7724 291] [278 7760]]	[[7739 276] [261 7777]]	[[7746 269] [260 7778]]	[[7760 255] [260 7778]]
test	[[1885 100] [100 1862]]	[[1900 85] [79 1883]]	[[1905 80] [76 1886]]	[[1910 75] [80 1882]]	[[1911 74] [81 1881]]

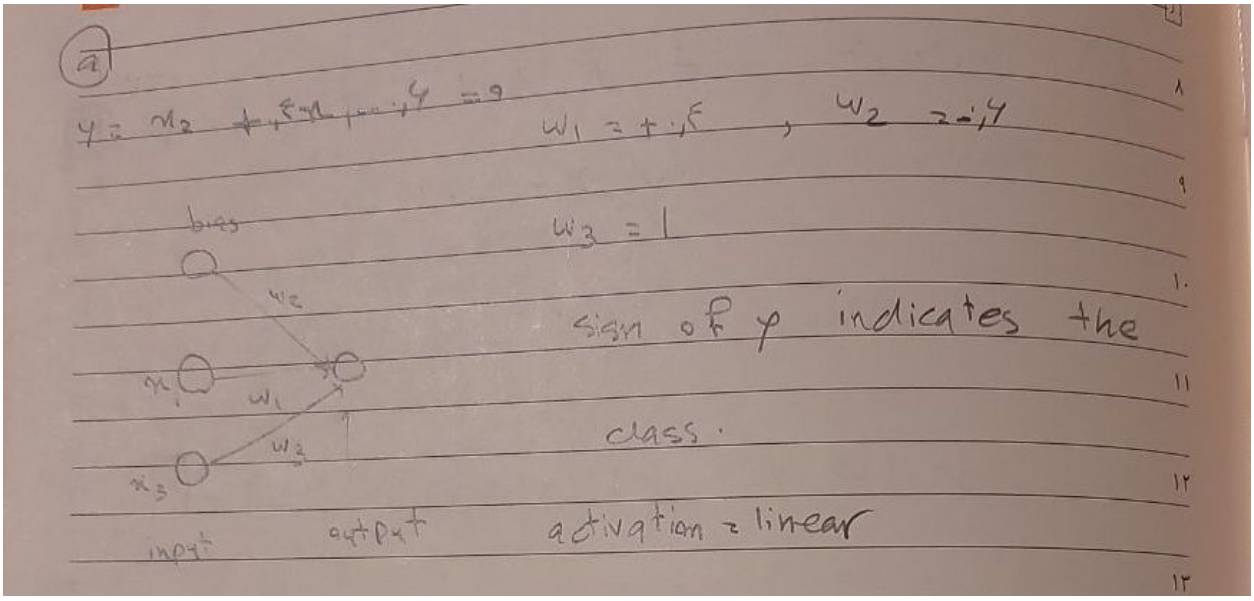
According to the results, the best K for KNN algorithms is 7 because it leads to a minimum test error.

Misclassified eyeglasses and pants are shown in the following table:

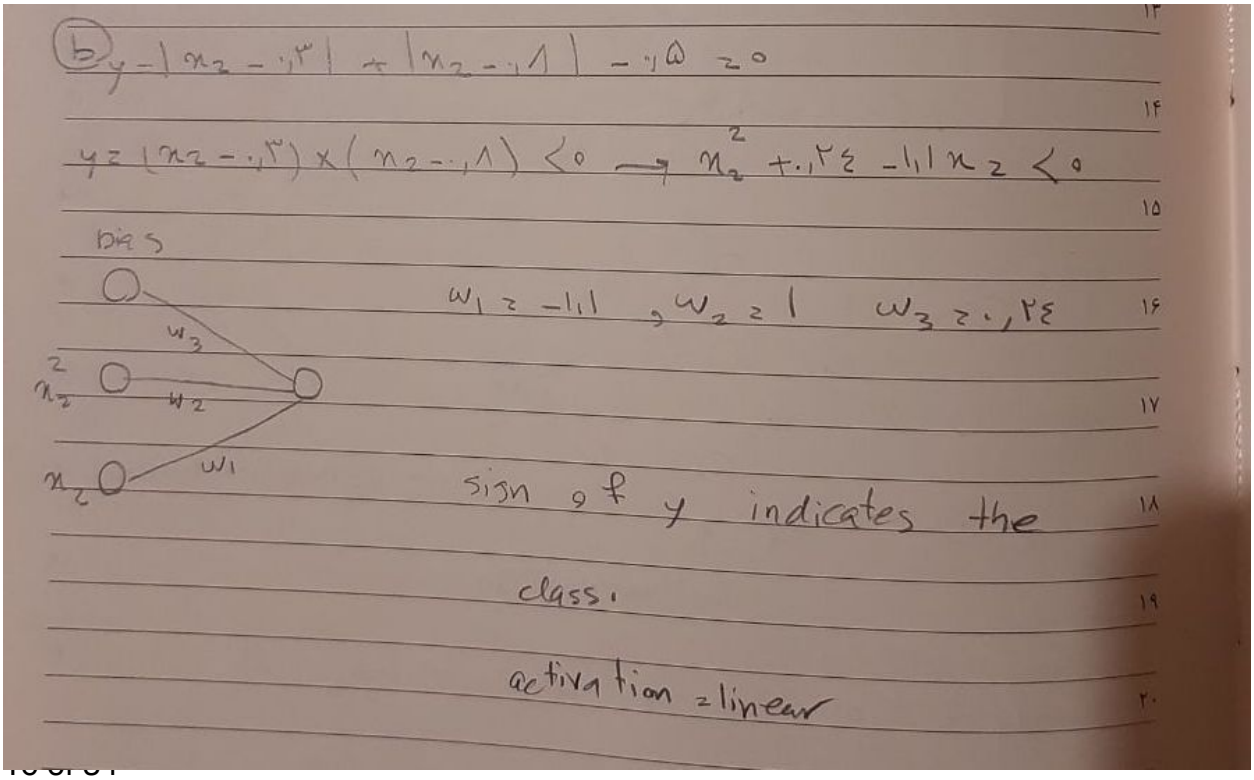
misclassified eyeglass	misclassified pant
	

question 4)

a)



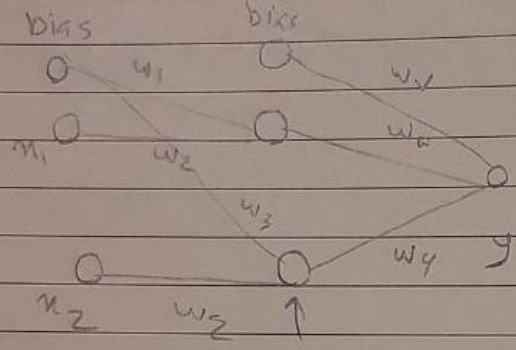
b)



b) $y = sign(x_2 - 0.3) + sign(0.1 - x_2) = ?$

if $y = 0 \rightarrow$ black

if $y < 0 \rightarrow$ white



$y \rightarrow$ activation = linear

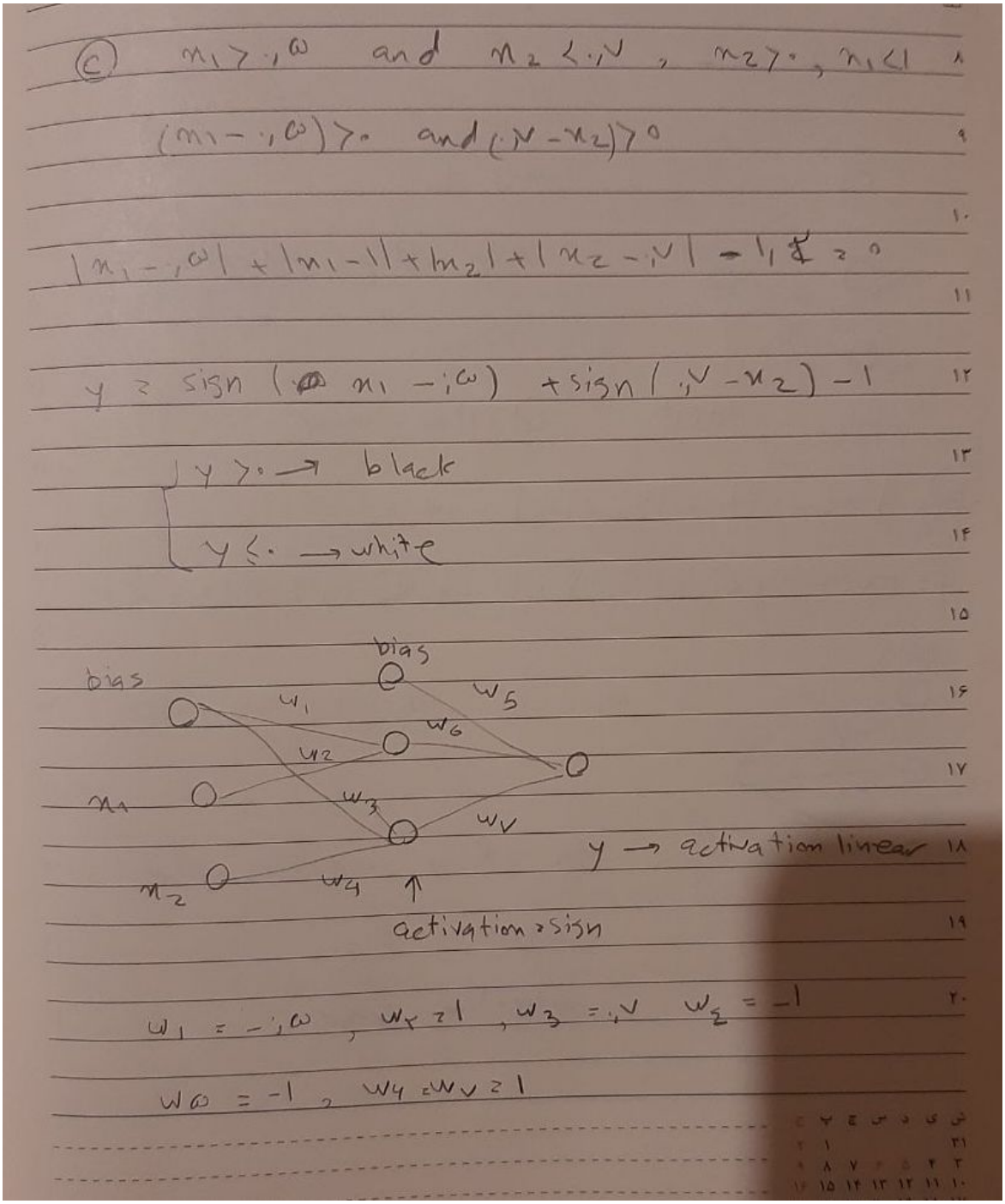
activation func = sign

$w_1 = -0.3$, $w_2 = 1$

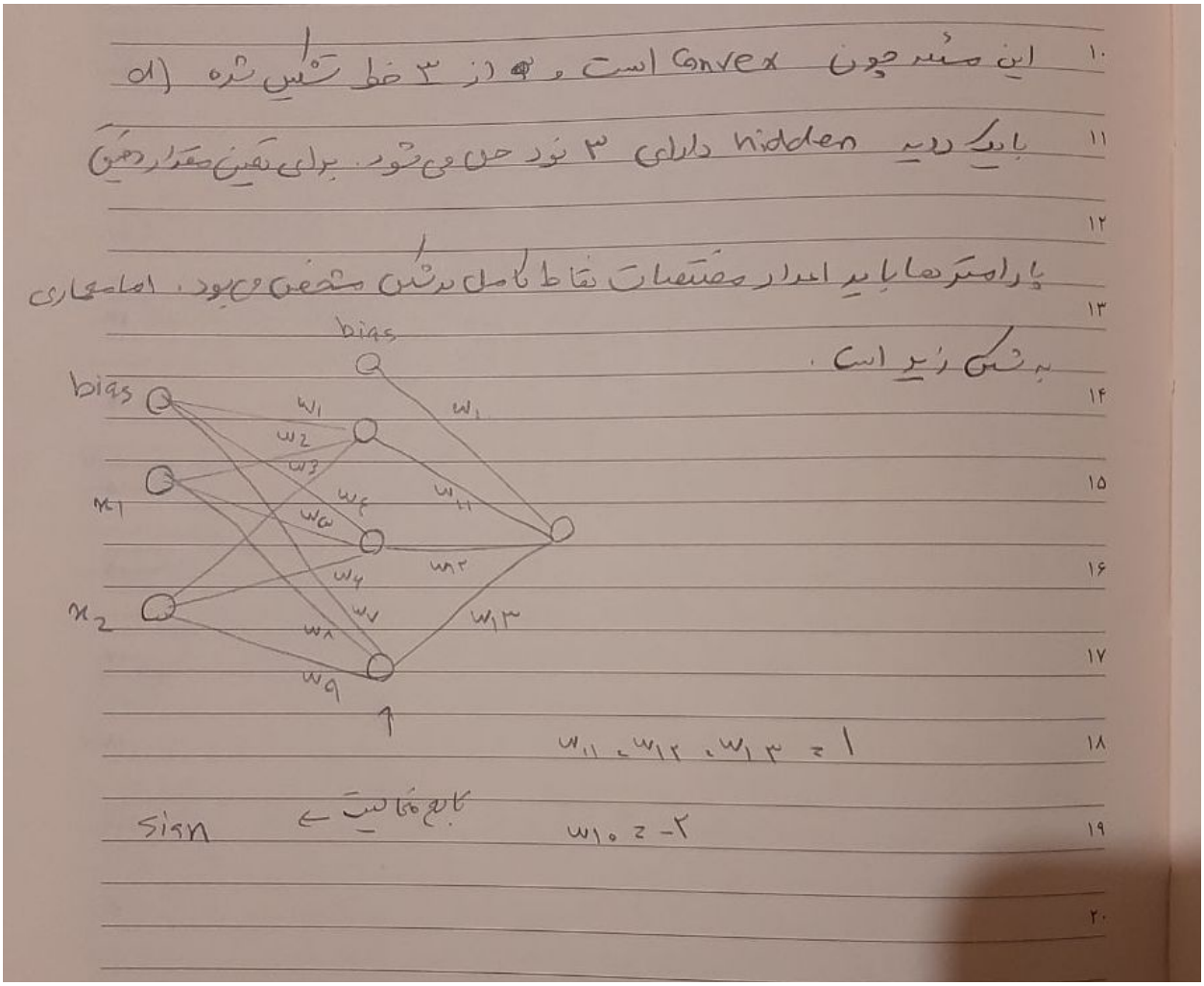
$w_3 = -1$, $w_4 = 0.1$

$w_5 = w_6 = 1$, $w_{12} = -1$

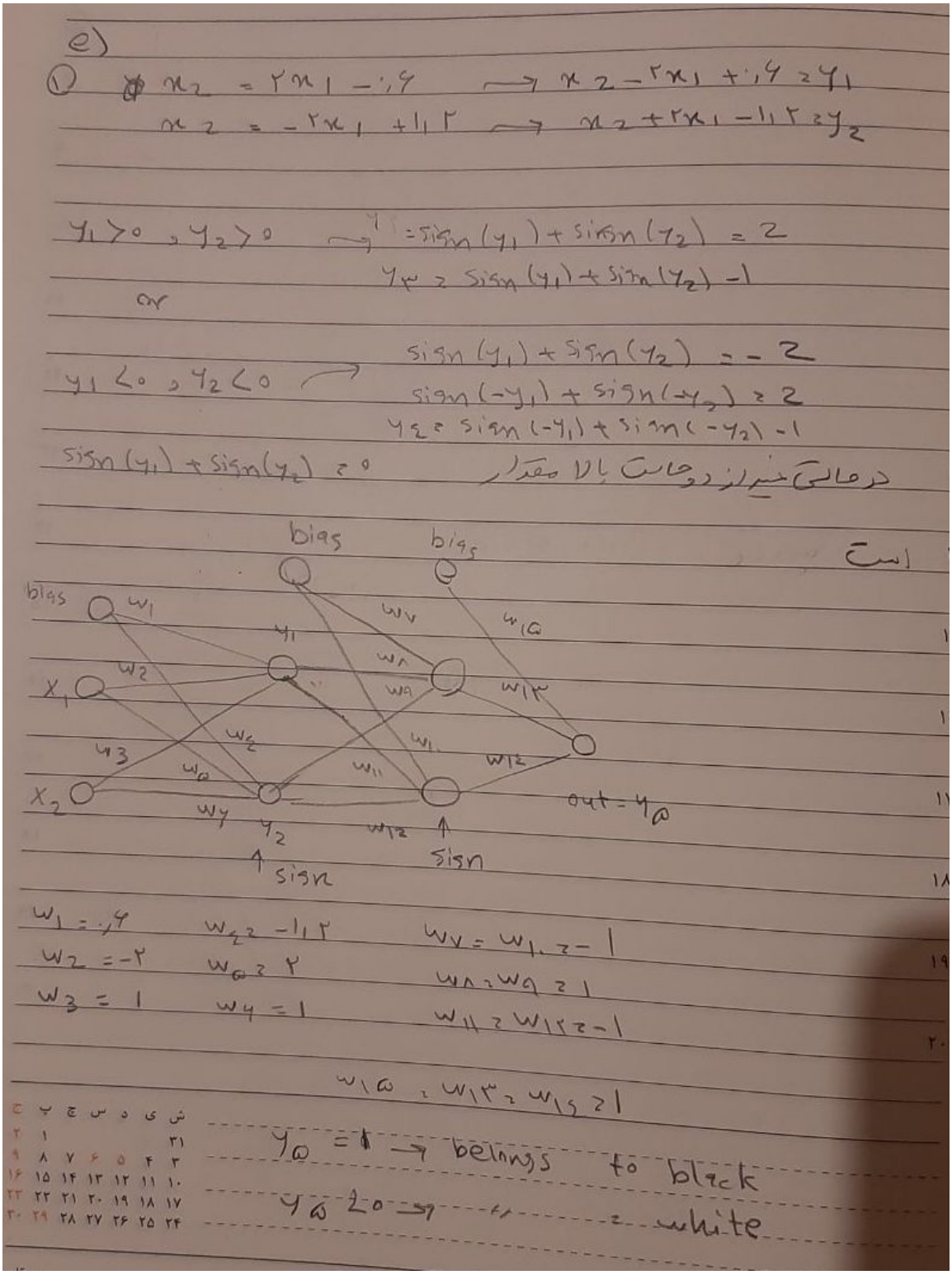
c)


c) $x_1 > 0.5$ and $x_2 < 0.5$, $x_2 > 0$, $x_1 < 1$
$(x_1 - 0.5) > 0$ and $(0.5 - x_2) > 0$
$|x_1 - 0.5| + |x_1 - 1| + |x_2| + |x_2 - 0.5| - 1.5 = 0$
$y = \text{sign}(x_1 - 0.5) + \text{sign}(0.5 - x_2) - 1$
$y > 0 \rightarrow$ black
$y \leq 0 \rightarrow$ white
bias
bias
w_1
w_2
w_3
w_4
w_5
w_6
w_7
x_1
x_2
$y \rightarrow$ activation linear
activation = sign
$w_1 = -0.5$, $w_2 = 1$, $w_3 = 0.5$ $w_4 = -1$
$w_0 = -1$, $w_4 = w_7 = 1$

d)


d) این مثلث چون Convex است، و از ۳ خط تشکیل شده
با یک دید hidden دارای ۳ نود حل می‌شود. برای تعیین مقدار وزن
پارامترها باید اندازه مختصات نقاط کامل بدانیم مشخص می‌شود. اما معماری
به شکل زیر است.
bias
bias
x_1
x_2
w_1
w_2
w_3
w_4
w_5
w_6
w_7
w_8
w_9
w_{11}
w_{12}
w_{13}
$w_1 = w_{11} = w_{12} = w_{13} = 1$
$w_{10} = -2$
تابع فعالیت ←
sign

e)


e)
① $x_2 = 2x_1 - 0.4 \rightarrow x_2 - 2x_1 + 0.4 = y_1$
$x_2 = -2x_1 + 1.2 \rightarrow x_2 + 2x_1 - 1.2 = y_2$
$y_1 > 0 , y_2 > 0 \rightarrow y = \text{sign}(y_1) + \text{sign}(y_2) = 2$
$y = \text{sign}(y_1) + \text{sign}(y_2) - 1$
or
$y_1 < 0 , y_2 < 0 \rightarrow \text{sign}(y_1) + \text{sign}(y_2) = -2$
$\text{sign}(-y_1) + \text{sign}(-y_2) = 2$
$y = \text{sign}(-y_1) + \text{sign}(-y_2) - 1$
در حالتی غیر از دو حالت بالا مقدار
$\text{sign}(y_1) + \text{sign}(y_2) = 0$
است
bias
bias
bias
x_1
x_2
y_1
y_2
sign
sign
out = y_5
$w_1 = 0.4$
$w_2 = -2$
$w_3 = 1$
$w_4 = 1$
$w_5 = -1.2$
$w_6 = 2$
$w_7 = w_{10} = -1$
$w_8 = w_9 = 1$
$w_{11} = w_{12} = -1$
$w_{13} = w_{14} = w_{15} = 1$
$y_5 = 1 \rightarrow$ belongs to black
$y_5 = 0 \rightarrow$ white

f) this network should at least has five following inputs: body height, body length, eye diameter, horn length and leg length. We can also extract better features from these features and give it as input to the network.

g) As this problem is a classification problem and there are 5 different classes in this problem, this network needs to have 5 outputs where each one indicates whether the given data belongs to that class or not.

h) Yes, this network should have at least one hidden network. Optimum count of hidden units would be determined by experiment. Actually deeper networks solve the problem better but it also depends on the size of the data. For training deep networks we need more data! We should also take into account over-fitting phenomena when determining the best depth.

I) Recurrent connections are useful when we have a series data, like a sequence of words, prices and etc. in this case we don't need recurrent connection and a deep feed forward network solves the problem.

J) Different activation functions like ReLu, logistic, TanH, sigmoid and softmax can be used for this problem, but usually softMax is used for multi-classification problem.
https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

k) Different learning algorithms like Adam, Adagrad, RMSProp, and SGD can be used for this problem.

question 5)

a)

```
38
39          v=b;
40          for j=1:nSV
41              v = v + alpha(j) * kernel_fnc(SV(j,:)',xy)
42          end
43
44          if (v < 0)
```

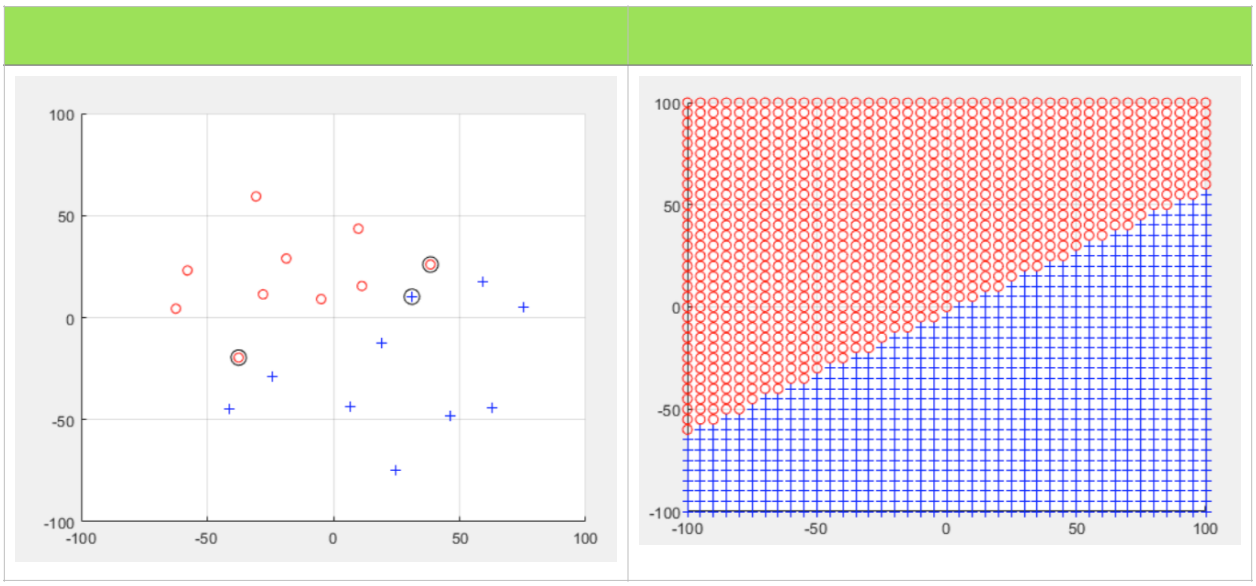
b)

```
42      for j=1:n
43
44          xj = D(j,1:2);
45          yj = D(j,3);
46
47          H(i,j) = kernel_fnc(xi',xj') * yj * yi ;
48
49      end
```

c)

```
75  options = optimset('LargeScale','off');
76
77  [x,fval] = quadprog(H, f, A, b, Aeq, beq, lb, ub, x0, options);
78
79  tol = 1e-8;
```

d)



outputs of this function are described below:
1) SV: is array of support vectors
2) alpha : is array of parameter alphas in SVM algorithm
3) b : is also parameter b in SVM

e) circles points indicate the support vectors.

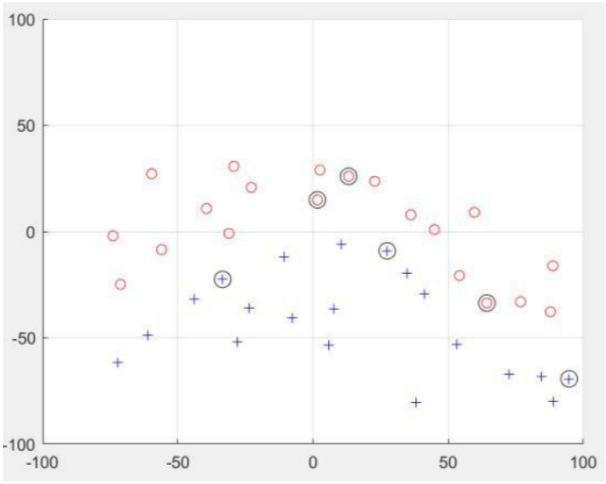
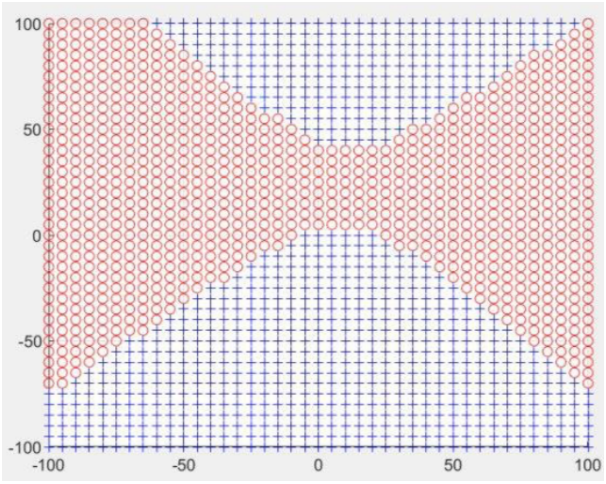
f) according to the comments in code, inf indicates cost of misclassifying. It is used for determining the cost of slack variable.

```
14  %  Format x_1, x_2, y  per line
15  %
16  %  C is the slack cost -- the cost of misclassifying  training point in
17  %   the non-separable case.  C = inf means separable case.
18
19
```

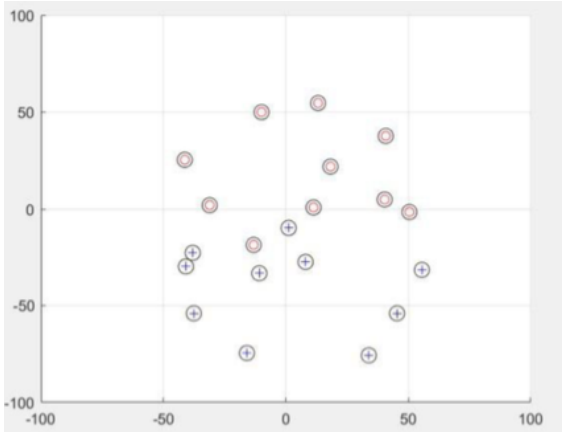
g) this new kernel function is as below:

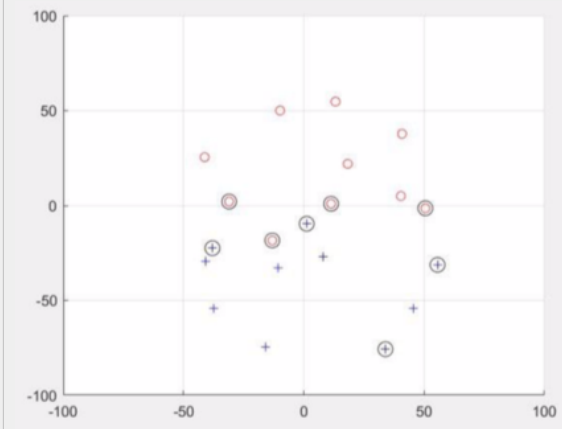
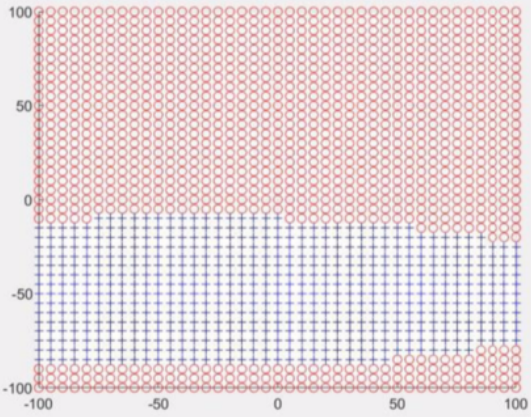
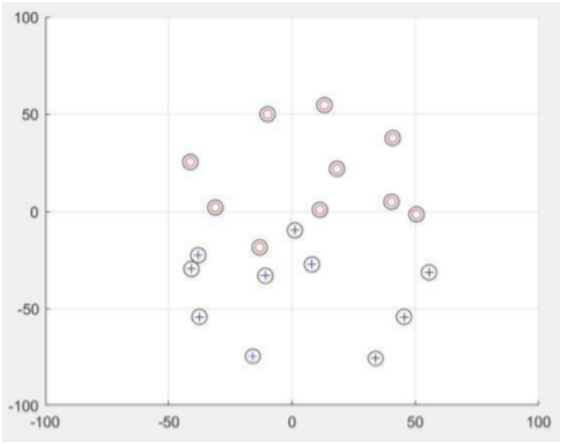
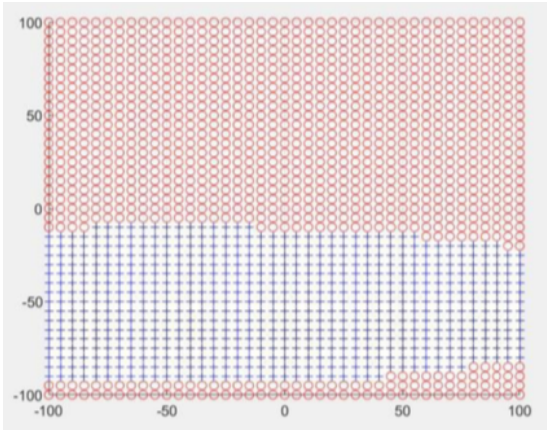
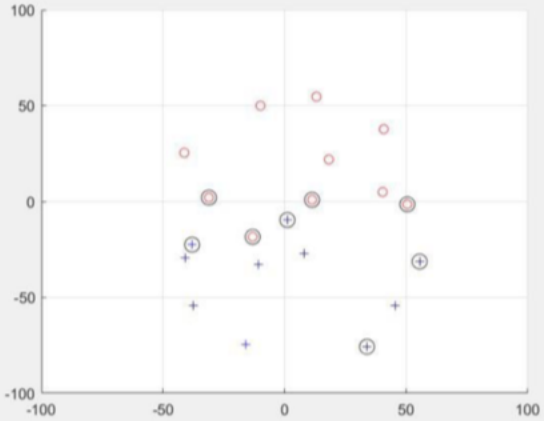
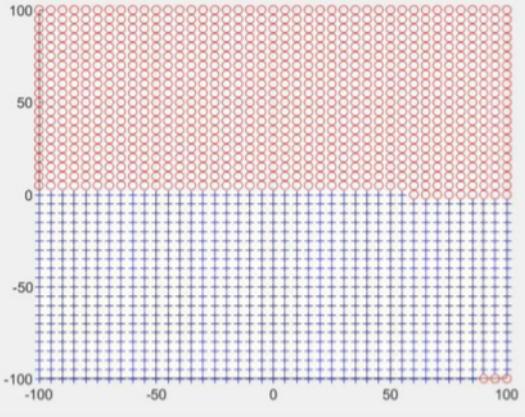
```
1  function p = quad_kernel(x,y,args)
2
3  p = (x'* y + 1 )^2;
4
```

result is shown in the following:

h)

inf param		
inf		

inf param		
10		
1		
0.1		

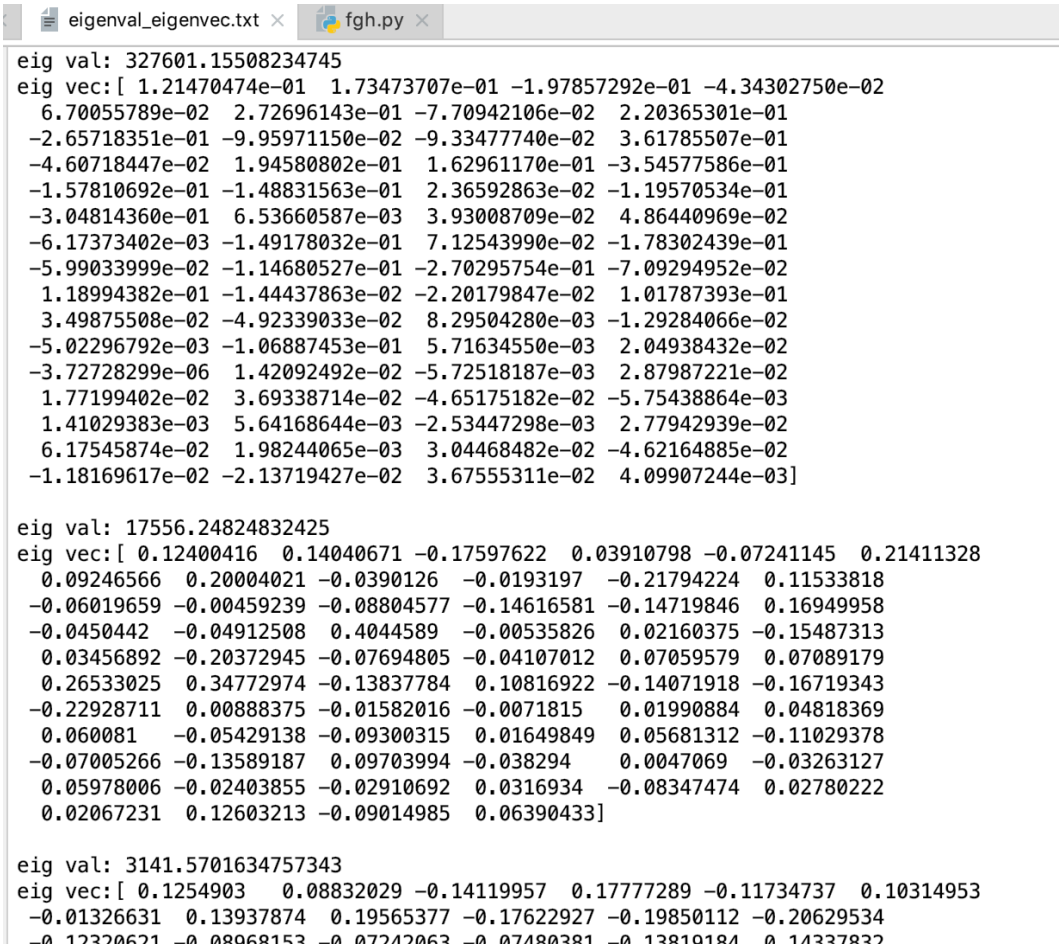
C is a regularization parameter that controls the trade off between the achieving a low training error and a low testing error that is the ability to generalize the classifier to unseen data.

Consider the objective function of a linear SVM : min |w|^2+C∑ξ. If C is too large the optimization algorithm will try to reduce |w| as much as possible leading to a hyperplane which tries to classify each training example correctly (similar to first figure). Doing this will lead to loss in generalization properties of the classifier. On the other hand if C is too small then you give your objective function a certain freedom to increase |w| a lot, which will lead to large training error.
C Parameter is used for controlling the outliers, low C implies we are allowing more outliers, high C implies we are allowing fewer outliers.

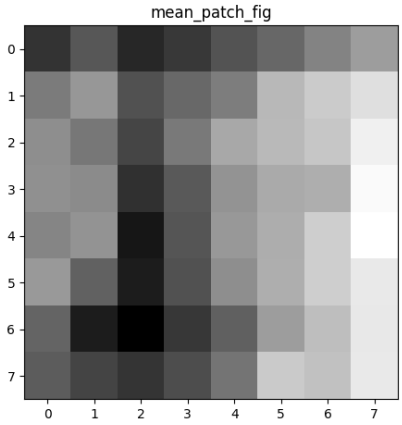
question 6)

A)

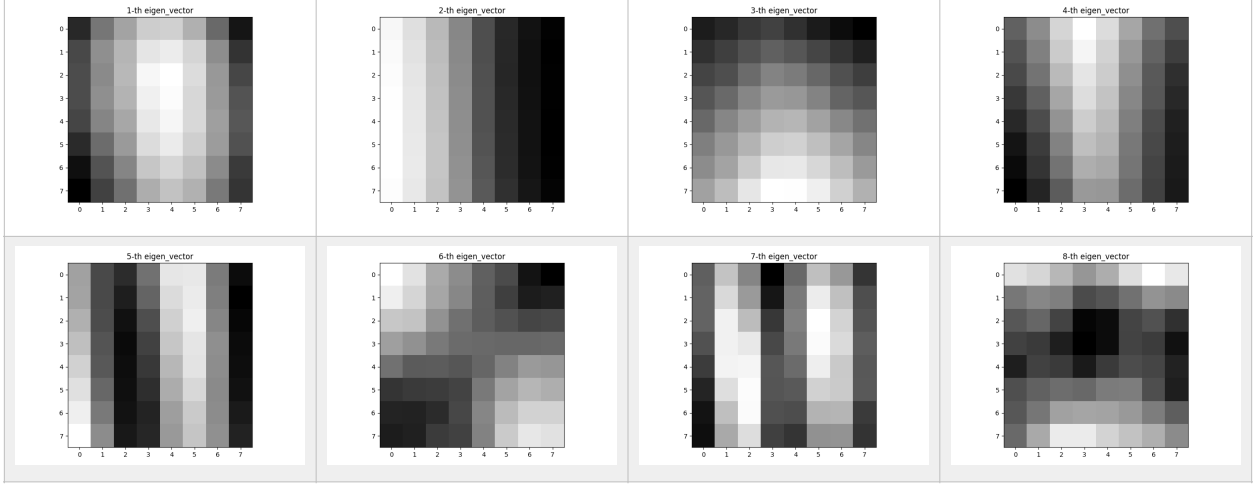
B) first 20 largest eigenvalues and corresponding eigenvectors are written in the ‘eigenval_eigenvec.txt’ file:



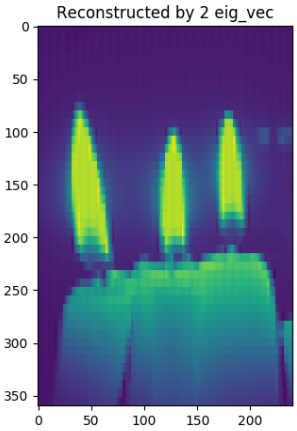
Here is the image of mean patch:

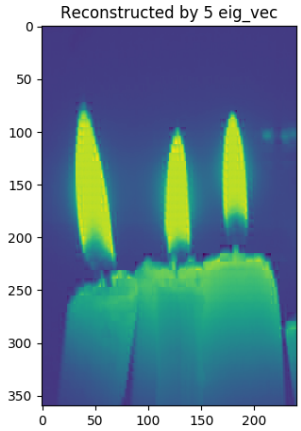


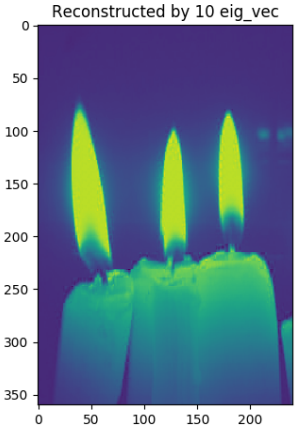
first 8 Eigen-vectors image

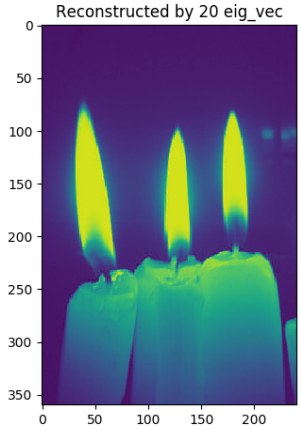


C, D, E)
Result is shown in the following table:









As we see by increasing the count of eigen-vectors quality of reconstructed image increases. Actually quality of the image is related to the variance ratio of used eigenvectors not number of used eigenvectors. based non the above figures quality doesn’t change more by adding 10 eigenvectors after first 10 eigenvectors.

F)
eigenvalues and eigenvectors of this question are written in file (eigenval_eigenvec_f.txt).

patch image:

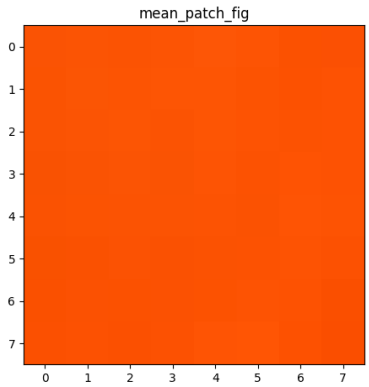
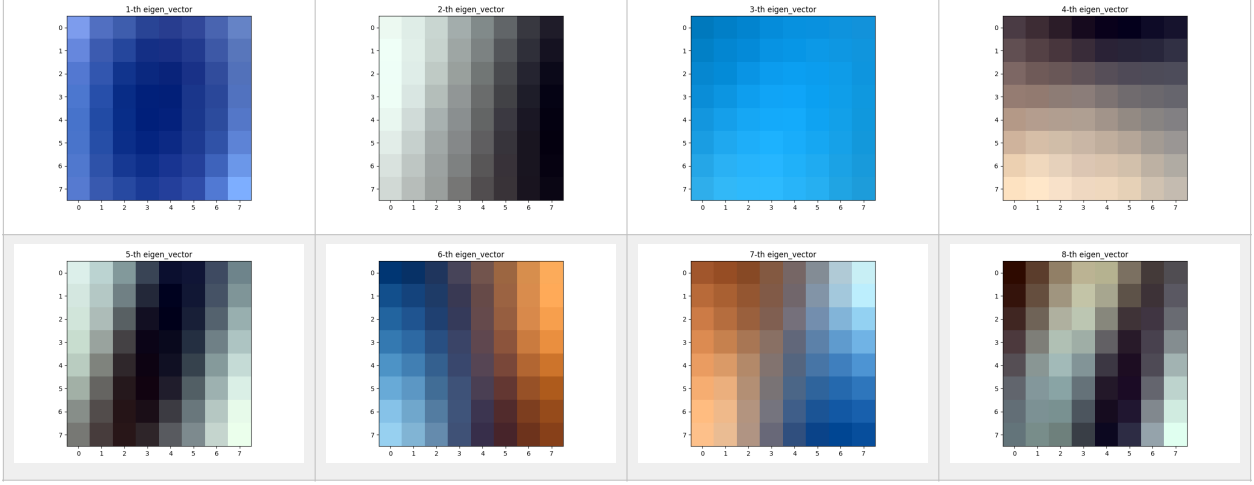
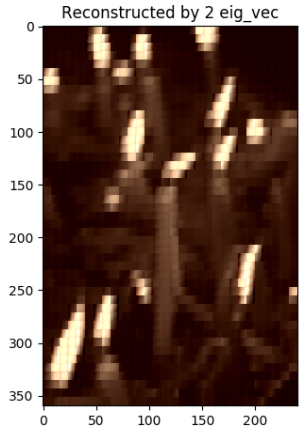
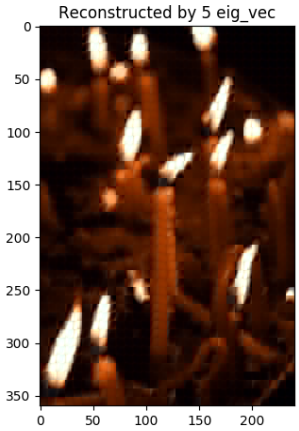


image of eigenvectors



Reconstructed images





question 7)

A. No it doesn't make sense, Because PCA is used for dimensionality reduction not for increase in dimensionality. also it works by extracting the eigenvectors and eigenvalues and it is not possible to extract Eigen vectors more than the size of dimensionality or size of the dataset.
B. consider two vectors W1 and W2. span(W1, W2) is zi1W1 + zi2W2 for scalars zi1 and zi2. in PCA we have label switching which means span(W1, W2) = span(W2, W1). Then PCA has no uniqueness because of the label switching which means we can exchange rows of matrix W with each other. changing rows of W means changing the order of factors which is possible. (https://www.cs.ubc.ca/~schmidtm/Courses/340-F17/L27.pdf) (W is the transformation matrix)
c) SVD converts the main matrix to three matrices like USV, but PCA eliminates less significant components and transforms data into a new space where properties are not correlated there. Relationship between SVD and PCA is describer below. (Ref: https://intoli.com/blog/pca-and-svd/)

Let the data matrix $\mathbf{X}$ be of n×p size, where n is the number of samples and p is the number of variables. Let us assume that it is centered, i.e. column means have been subtracted and are now equal to zero.
Then the p×p covariance matrix $\mathbf{C}$ is given by $\mathrm{C} = \mathrm{X}^{\mathrm{T}}\mathrm{X}/(\mathrm{n}-1)$. It is a symmetric matrix and so it can be diagonalized:

$$\mathrm{C} = \mathrm{VLV}^{\mathrm{T}}$$

where $\mathbf{V}$ is a matrix of eigenvectors (each column is an eigenvector) and $\mathbf{L}$ is a diagonal matrix with eigenvalues λi in the decreasing order on the diagonal. The eigenvectors are called *principal axes* or *principal directions* of the data. Projections of the data on the principal axes are called *principal components*, also known as *PC scores*; these can be seen as new, transformed, variables. The j-th principal component is given by j-th column of $\mathbf{XV}$. The coordinates of the i-th data point in the new PC space are given by the i-th row of $\mathbf{XV}$.
If we now perform singular value decomposition of $\mathbf{X}$, we obtain a decomposition

$$\mathrm{X} = \mathrm{USV}^{\mathrm{T}}$$

where $\mathbf{U}$ is a unitary matrix and $\mathbf{S}$ is the diagonal matrix of singular values si. From here one can easily see that:

$$\mathrm{C} = \mathrm{V}\frac{\mathrm{S}^2}{n-1}\mathrm{V}^{\mathrm{T}}$$

meaning that right singular vectors $\mathbf{V}$ are principal directions and that singular values are related to the eigenvalues of covariance matrix via $\lambda i = s2i/(n-1)$. Principal components are given by $\mathbf{XV}=\mathbf{US}V^T\mathbf{V}=\mathbf{US}$.
To summarize:

1. If $\mathbf{X}=\mathbf{US}\mathrm{V}^T$, then columns of $\mathbf{V}$ are principal directions/axes.
2. Columns of $\mathbf{US}$ are principal components ("scores").
3. Singular values are related to the eigenvalues of covariance matrix via $\lambda i = s2i/(n-1)$. Eigenvalues λi show variances of the respective PCs.
4. To reduce the dimensionality of the data from p to $k<p$. select k first columns of $\mathbf{U}$, and $k\times k$ upper-left part of $\mathbf{S}$. Their product $\mathbf{U}k\mathbf{S}k$ is the required $n\times k$ matrix containing first k PCs.

d) We know that PCA ignores small values of Eigen-values which leads to numerical instability. As singular values are more numerical stable than eigenvalues, in this case it would be better to use SVD for performing PCA instead of calculating the XX and performing Eigen-value and Eigen-vector decomposition. In the previous part it is explained completely how to use SVD for performing PCA. But here is a summary:
SVD on matrix X results in obtaining matrices U, S, and V as:

$$X = USV$$

where columns of US are principal components.
using SVD may take longer time but it has higher numerical accuracy.

e) PCA can reduce noise but it can't eliminate the noise. In PCA an orthogonal linear transformation is used to find a projection of all data into k dimensions, whereas these k dimensions are those of the highest variance. The eigenvectors of the covariance matrix (of the dataset) are the target dimensions and they can be ranked according to their eigenvalues. A high eigenvalue signifies high variance explained by the associated eigenvector dimension. This process of keeping large eigenvalues and corresponding eigenvectors helps to reduce the noise and keep the most meaningful information of image.

f) Back Propagation is used to effectively train a neural network through a method called chain rule. In simple terms, after each forward pass through a network, back propagation performs a backward pass while adjusting the model's parameters (weights and biases). if the used activation function in network is not differentiable then the BP is not applicable.
networks like restricted boatsman machine (RBM) and Hard attention can't use BP.

g) yes it is possible. It is differentiable and has returns a value between 0 and 1. but it is not used in deep neural networks because of following problems:

1) It kills gradients at saturation point, and cause vanishing gradient problem,
2) Its outputs is not zero-centered.

h) yes it can be used for regression. SVM works based on maximizing the margin. Maximizing the margin can more generally be seen as regularizing the solution by minimizing W (which is essentially minimizing model complexity) this is done both in the classification and regression. But in the case of classification this minimization is done under the condition that all examples are classified correctly and in the case of

regression under the condition that the value Y of all examples deviates less than the required accuracy ϵ from f(x) for regression.

In Regression the goal is to find a function f(x) = wx+b under the condition that f(x) is within a required accuracy ϵ from the value value y(x) of every data point, i.e. |y(x) –f(x)| ≤ ϵ where epsilon is the distance between the red and the grey line. Under this condition we again want to minimize f′(x)=w, again for the reason of regularization and to obtain a unique solution as the result of the convex optimization problem. One can see how minimizing w results in a more general case as the extreme value of w=0 would mean no functional relation at all which is the most general result one can obtain from the data.