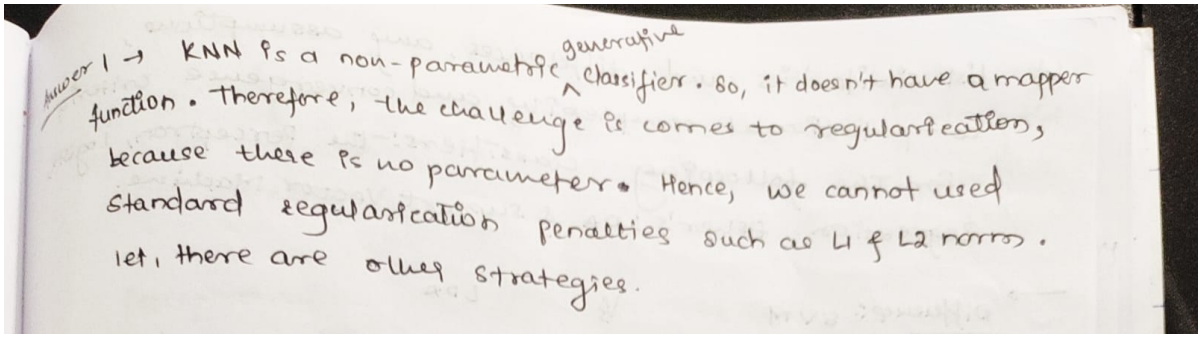
Homework 3 Part 1

This is an individual assignment.

Answer the following questions in markdown cells.

Problem 1 (4 points)

Suppose you are performing classification using the k-NN algorithm. Would you be able to apply regularization with lasso or ridge regularization? Why or why not?



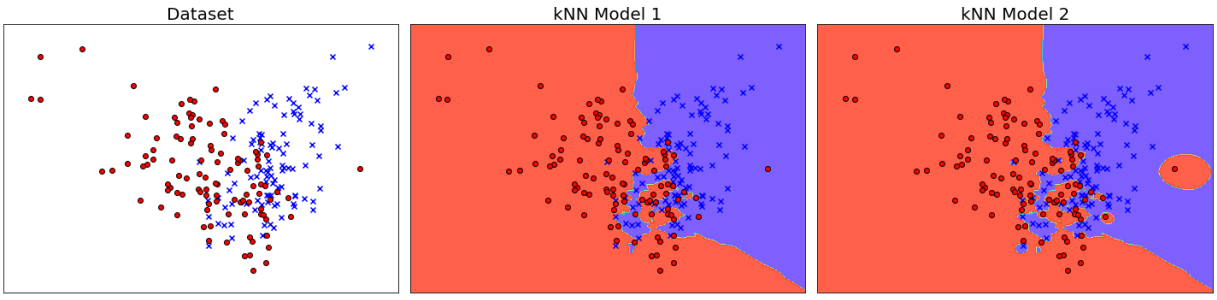
In [ ]:

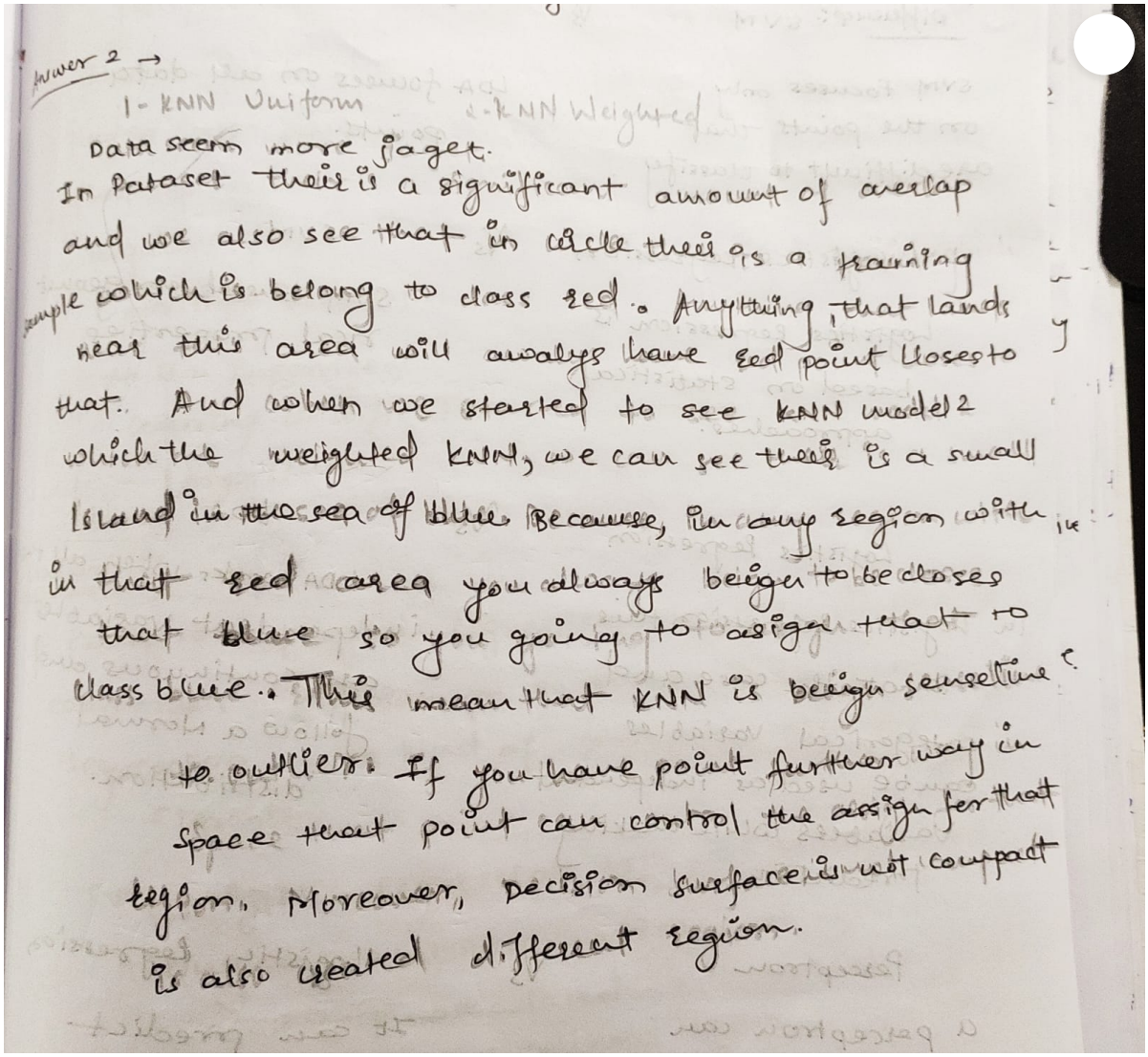
Problem 2 (4 points)

Consider the following two-dimensional dataset with two classes (shown with "circles" and "crosses"), and its decision surface performance as computed using k-Nearest Neighbor (kNN) algorithm with $k = 3$. Describe in words the major differences you observe in the two decision surfaces, and provide a discussion about the voting system utilized to obtain each respective decision surface. Justify your answer.

In [1]:
```
from IPython.display import Image
Image('figures/kNN-performances.png',width=900)
```

Out[1]:





In [ ]:

Problem 3 (5 points)

Recall that the Support Vector Machine objective function is

$$C\sum_{n=1}^{N}\xi_n + \frac{1}{2}\|w\|^2$$

subject to the constraints

$$t_n y(x_n) \geq 1 - \xi_n, n = 1, \ldots, N \quad (1)$$
$$\xi_n \geq 0, n = 1, \ldots, N \quad (2)$$

Answer the following questions:

1. **From the training set, which points are used to make predictions during the test stage? Explain your reasoning.**
2. **C is a parameter that is set by the user. Describe the relationship between values of C and the resulting SVM decision surface, performance and number of support vectors.**

Problem 3 Answer

1) Discriminant function: $y(x) = w^T \phi(x) + b$

$$y(x) = \sum_{n=1}^{N} a_n t_n k(x_n, x) + b$$

$a_n = 0$ for correctly classified

$\& a_n \neq 0$ for misclassified

for b (we know that the S.V.S x_n, satisfy $t_n \cdot y(x_n) = 1$

$(\xi_n = 0)$ $b = \frac{1}{N_M} \sum_{n \in M} \left(t_n - \sum_{m \in S} a_m t_m k(x_n, x_m) \right)$

- Although predictions for new points/inputs are made using only the support vectors, the training phase (i.e the determination of the parameter $a \& b$) makes use of the whole dataset, and so it is important to have efficient algorithms for solving the quadratic programming problem. the objective funct^n $L(a)$ is quadratic and so any local optimum will also be a global optimum provided the constraint.

2) If C is larger, then slack variable go to zero the margine will be very very small. If C is converges to zero, going to all more points enter to margin then we came with wider margine.

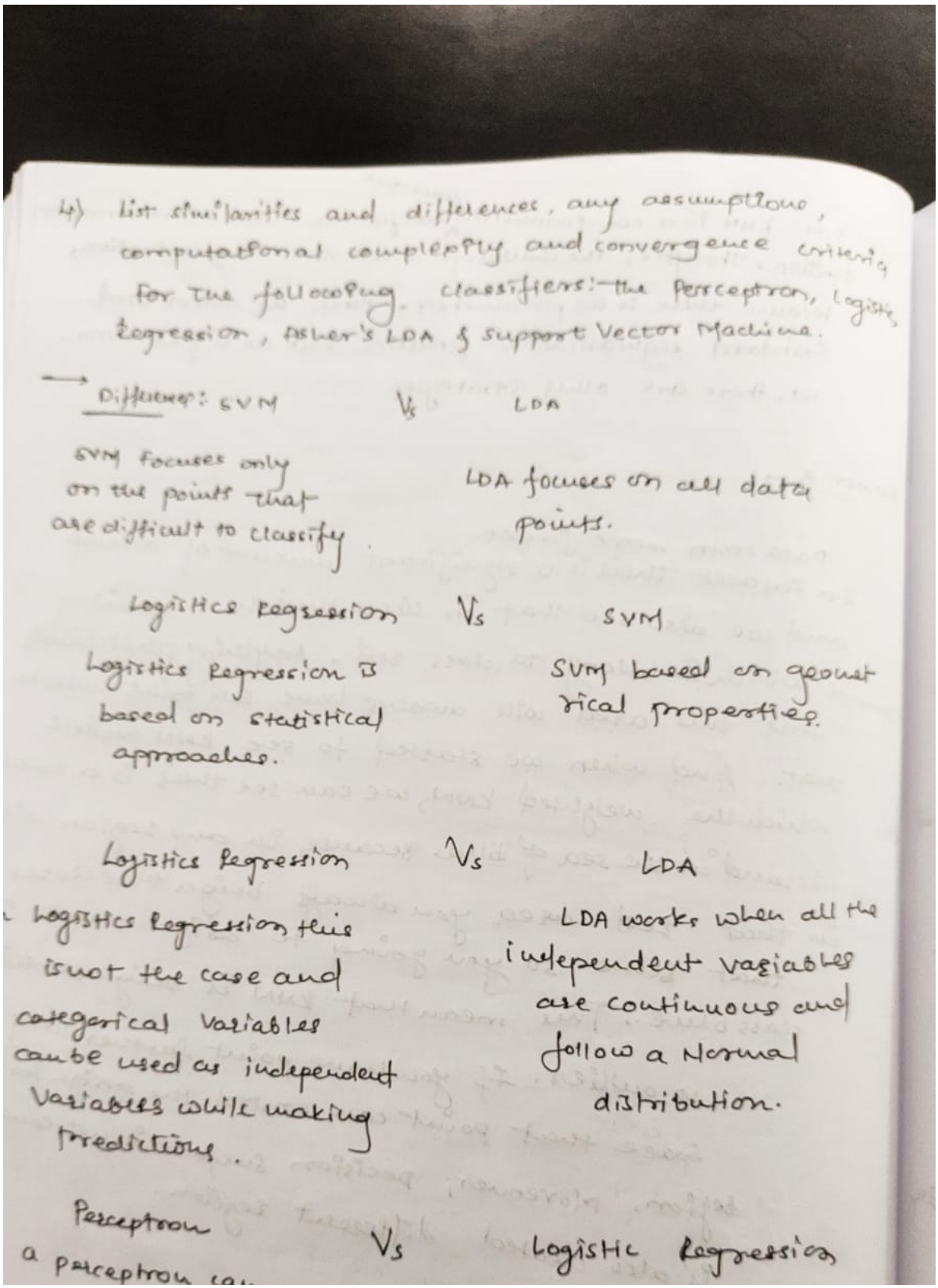
- C control that how many points enters the margine.

- The optimal value for the 'C' parameter will depend on your data set, and should be tuned using cross-validation or a similar procedure.

In [ ]:

Problem 4 (5 points)

List similarities and differences, any assumptions, computational complexity and convergence criteria for the following classifiers: the Perceptron, Logistic Regression, Fisher's LDA and Support Vector Machine (SVM).



4) List similarities and differences, any assumptions, computational complexity and convergence criteria for the following classifiers: the Perceptron, Logistic Regression, Fisher's LDA & Support Vector Machine.

→ Difference: SVM Vs LDA

SVM focuses only on the points that are difficult to classify.

LDA focuses on all data points.

Logistics Regression Vs SVM

Logistics Regression is based on statistical approaches.

SVM based on geometrical properties.

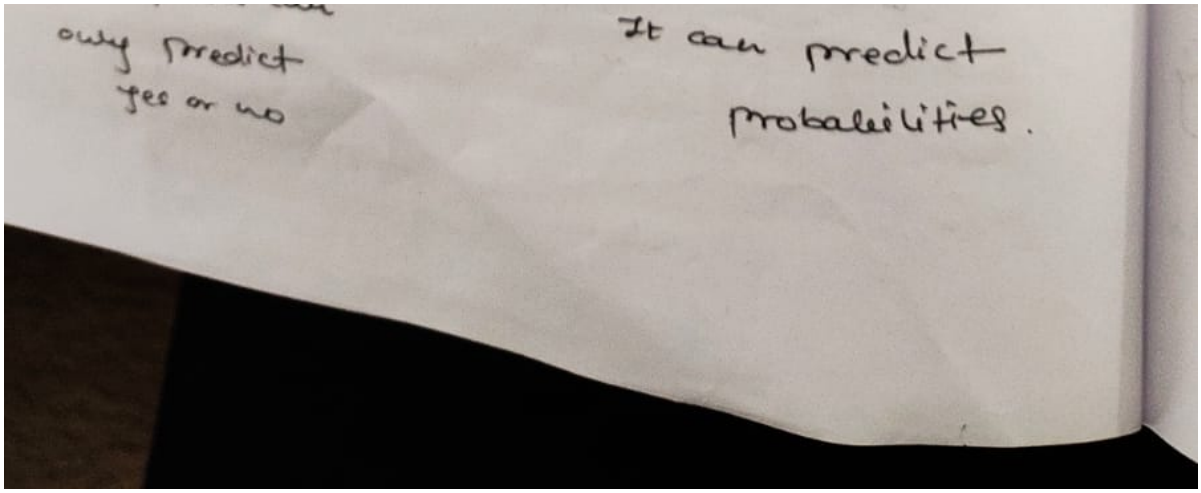
Logistics Regression Vs LDA

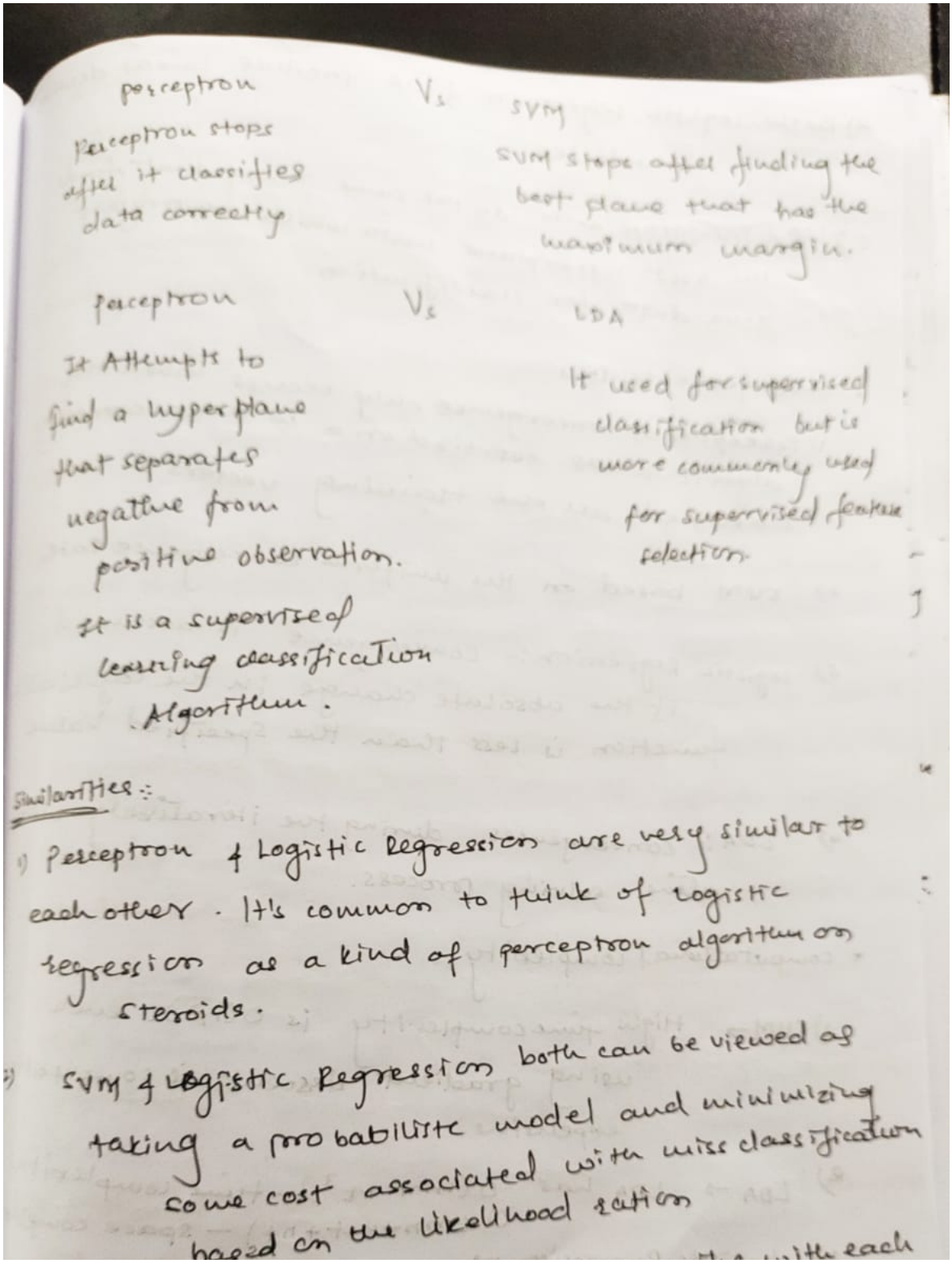
Logistics Regression this is not the case and categorical variables can be used as independent variables while making predictions.

LDA works when all the independent variables are continuous and follow a Normal distribution.

Perceptron Vs Logistic Regression

a perceptron


only predict
yes or no
It can predict
probabilities.


perceptron
Vs
SVM
Perceptron stops after it classifies data correctly
SVM stops after finding the best plane that has the maximum margin.
perceptron
Vs
LDA
It Attempts to find a hyperplane that separates negative from positive observation.
It is a supervised learning classification Algorithm.
It used for supervised classification but is more commonly used for supervised feature selection.
Similarities :-
1) Perceptron & Logistic Regression are very similar to each other. It's common to think of logistic regression as a kind of perceptron algorithm on steroids.
2) SVM & Logistic Regression both can be viewed as taking a probabilistic model and minimizing some cost associated with miss classification based on the likelihood ration

If in case two classes merged heavily with other then LDA & SVM could likely give same prediction.

4) Both logistic regression & LDA produce linear decision boundaries.

5) LDA & Perceptron both do the same thing that finds the best hyperplane unto which to project your data for classification.

- convergence criteria

1) Perceptron:- convergence only occurs when the algorithm has set-tied on a w that correctly classified all the training vectors in T.

2) SVM based on the uniform convergence rate,

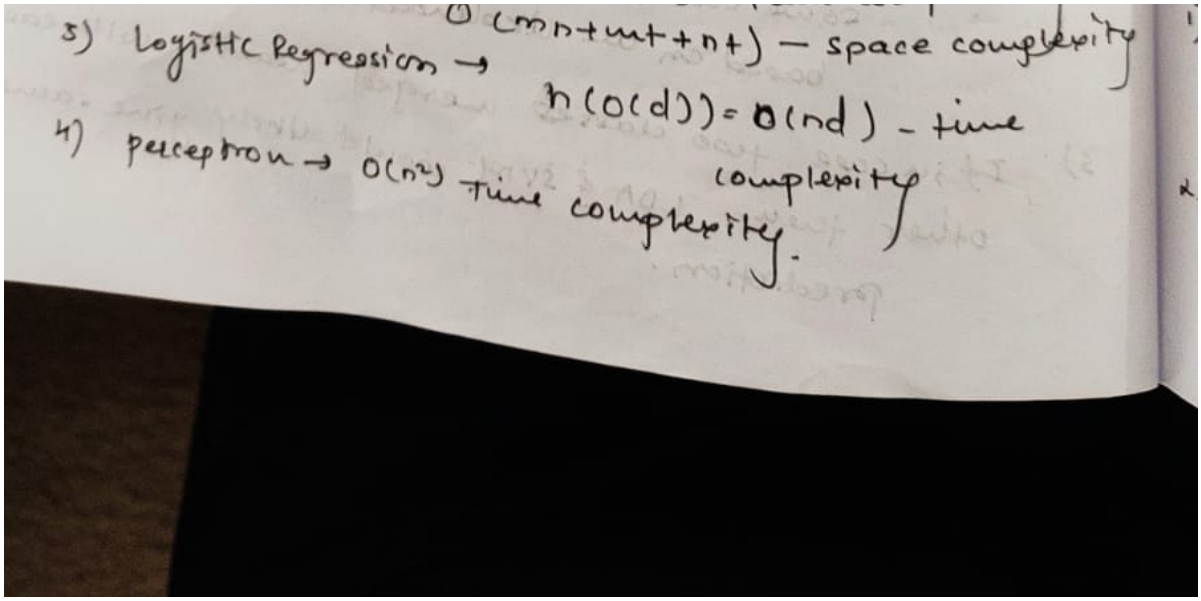
3) logistic Regression:- convergence is assumed if the absolute change in the log-likelihood function is less than the specified value.

4) LDA:- convergences during the iteratively alternative solving process.

- computational complexity:-

1) SVM → High time complexity is O(n3). even using gradient descent is computationally expensive.

2) LDA → LDA has O(mnt + t³) - time complexity



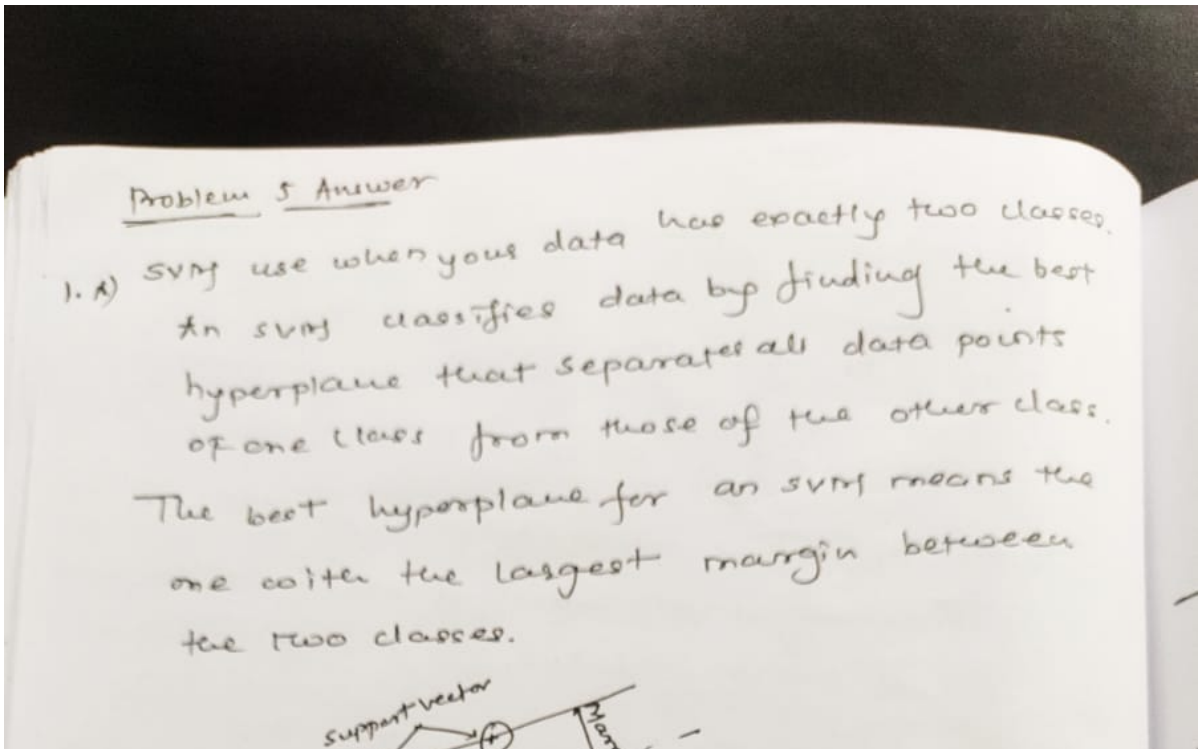
In [ ]:

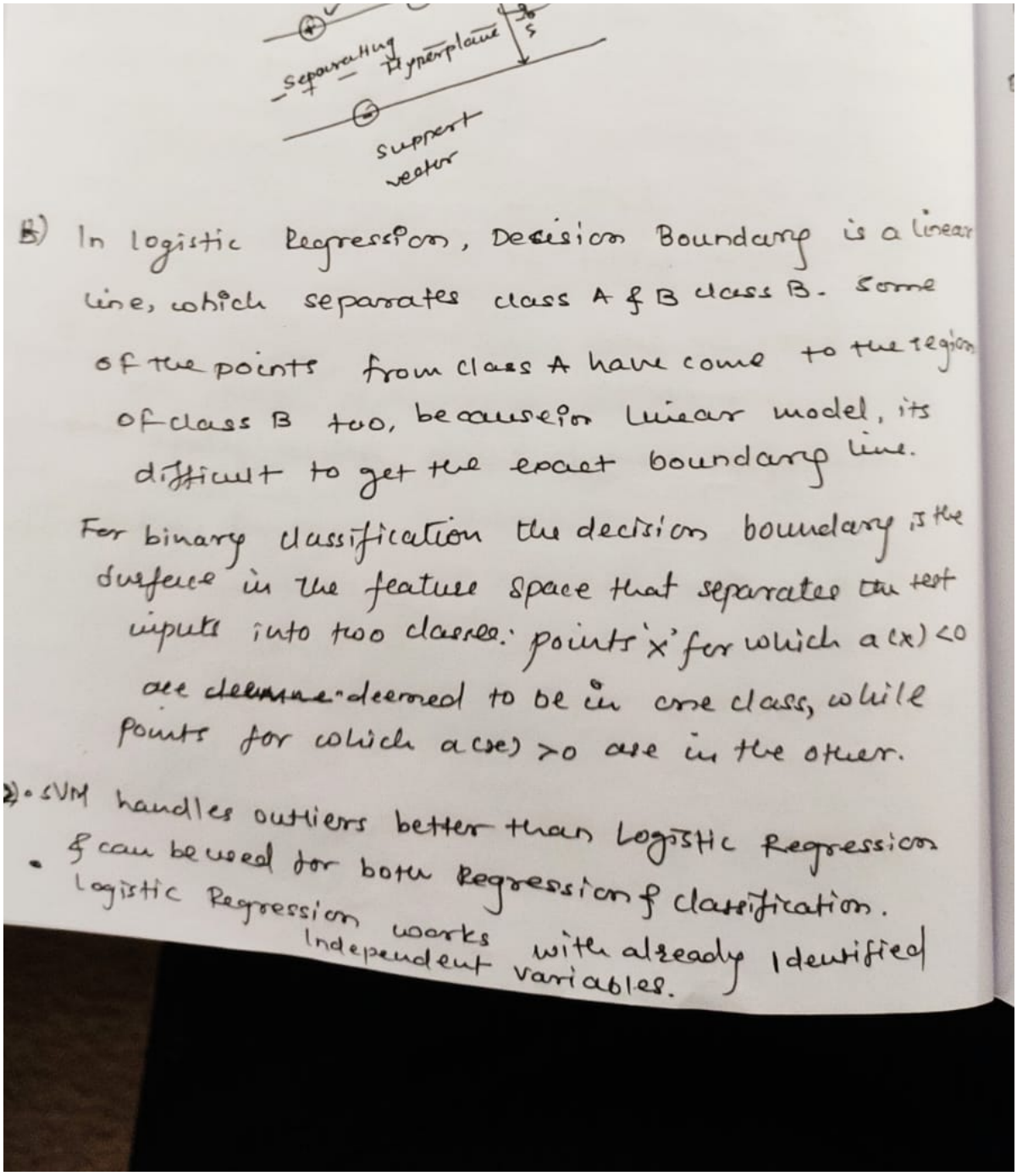
In [ ]:

Problem 5 (5 points)

Suppose you have a binary (2-class) classification problem. Answer the following questions:

1. **Describe how the logistic regression and the SVM find a decision surface to classify the two classes.**
2. **When would you choose SVM over Logistic Regression and vice-versa? Justify your answers.**





In [ ]:

Problem 6 (5 points)

Suppose you have the following training labels associated with binary classifier:

$$t = \{1, 1, 0, 0, 1, 0, 0, 1, 1, 0\}$$

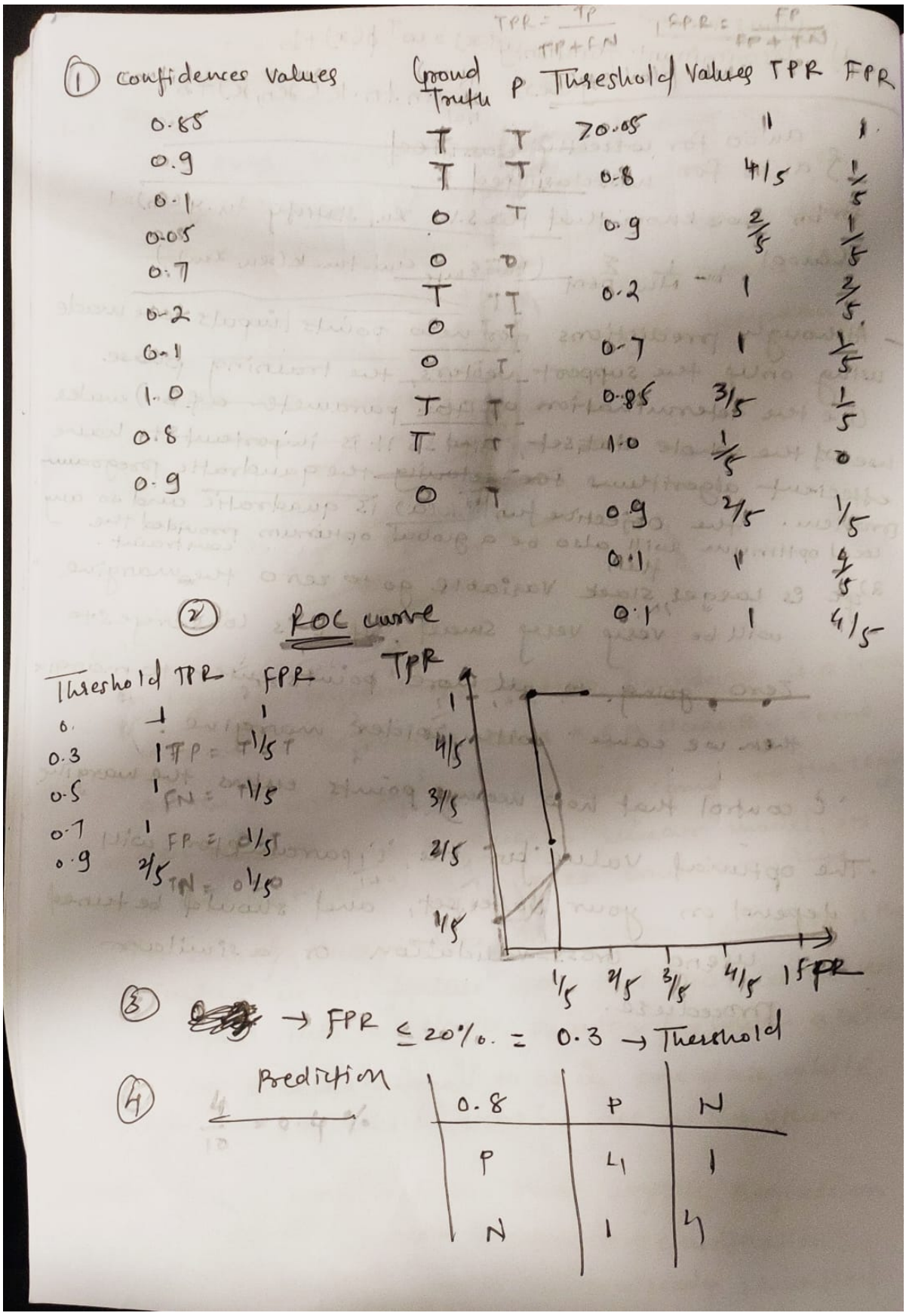
Suppose you trained a logistic regression classifier to produce a confidence of target given a sample. For the above data points, your classifier produced the following confidence values:

$$c = \{0.85, 0.9, 0.1, 0.05, 0.7, 0.2, 0.1, 1.0, 0.8, 0.9\}$$

the corresponding weighted sum $z = \mathbf{w}^T\mathbf{x} + w_0$ are:

$$z = \{4, 5, -5, -6, 1, -1.5, -1, -1, 4.5, 5.5\}$$

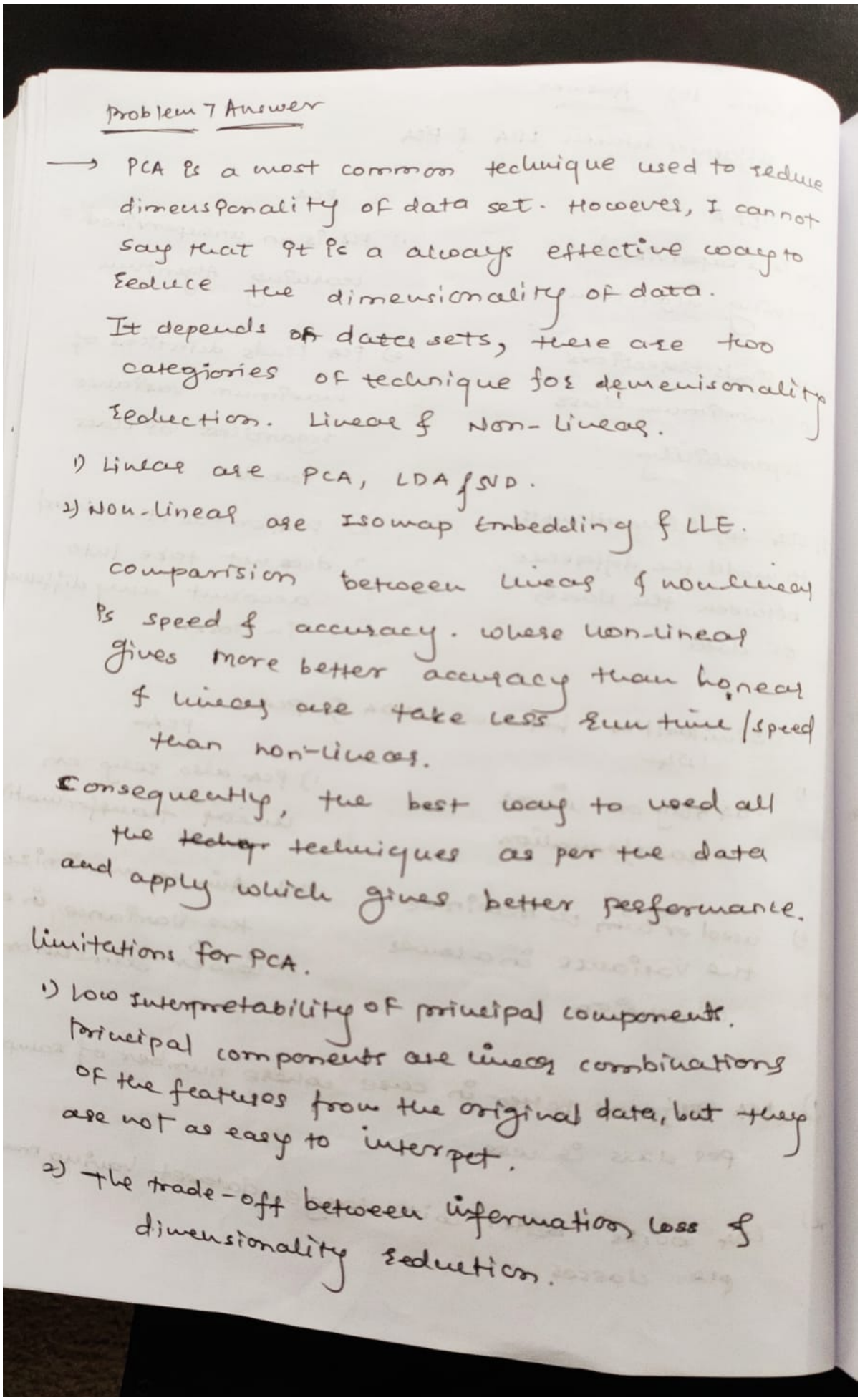
1. **Create a table with several entries (at least 10) that computes TPR and FPR for a list of possible thresholds for the confidence c, δ.**
2. **From the table you computed in 1, draw the associated ROC curve.**
3. **Which threshold would you use to achieve a FPR ≤ 20%?**
4. **Suppose you decided your threshold point of operation is ≥ 0.8. What is the resulting confusion matrix?**



In [ ]:

Problem 7 (4.5 points)

Suppose you would like to use PCA to reduce dimensionality of your data prior to classification. Is PCA always an effective dimensionality reduction technique to be used in conjunction with classification? Why or why not?



Problem 7 Answer

→ PCA is a most common technique used to reduce dimensionality of data set. However, I cannot say that it is a always effective way to reduce the dimensionality of data.
It depends on data sets, there are two categories of technique for dimensionality reduction. Linear & Non-linear.

1) Linear are PCA, LDA & SVD.
2) Non-linear are Isomap Embedding & LLE.

comparision between linear & nonlinear is speed & accuracy. where Non-linear gives more better accuracy than linear & linear are take less run time / speed than non-linear.

Consequently, the best way to used all the ~~techn~~ techniques as per the data and apply which gives better performance.

limitations for PCA.

1) Low interpretability of principal components.
Principal components are linear combinations of the features from the original data, but they are not as easy to interpret.
2) the trade-off between information loss & dimensionality reduction.

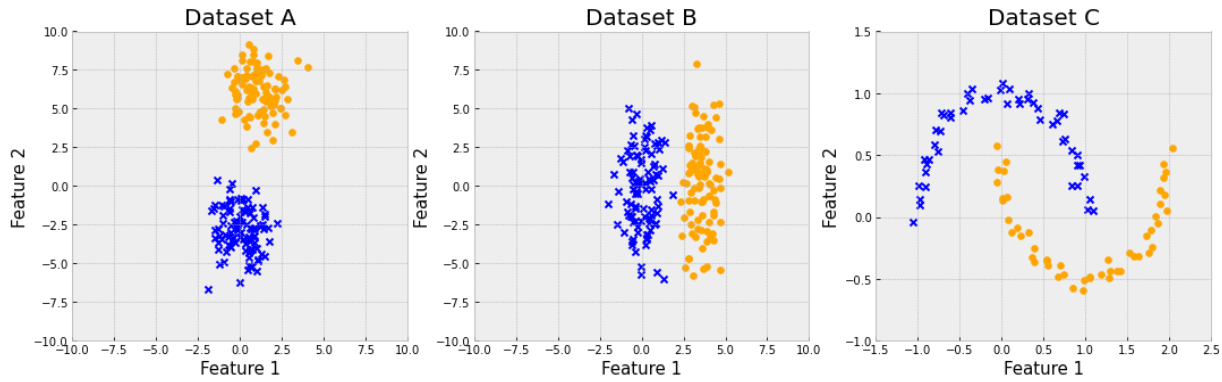
In [ ]:

Problem 8 (5 points)

Consider the following three two-dimensional datasets containing two classes (depicted as "circles" and "crosses").

In [2]:
```
Image('figures/2-D-datasets.png',width=800)
```

Out[2]:



Suppose you would like to apply Principal Component Analysis (PCA) to reduce the dimensionality of each of these datasets from 2-D to 1-D where the two clusters remain separated in the projection space. For each dataset (A, B and C), address each of the following two questions:

1. **Will PCA be effective at keeping the two clusters separated in the 1-D projection? Why or why not? If yes, state what characteristics of the dataset allow PCA to be effective. If no, state what characteristics of the dataset cause PCA to fail.**

2. **Can you think of another dimensionality reduction technique that would be successful at reducing the dimensionality for this dataset while maintaining (or increase) class separability? State the other method and describe why it would be successful.**

Answer

1) No, reducing dimensionality with PCA will only maximize variance, which may or may not translate to linear separability.

Here are two visualizations of variance and separability in opposition. In both cases, the discriminative information lies primarily along the low-variance axis, which would get discarded by rote dimensionality reduction

For dataset A B and C: Applying PCA on all of these dataset helps to reduce the dimesionality from 2-D to 1-D, such as the axis that explains the maximum amount of variance in the training

set is called pricipal components and the axis orthogonal to this axis is called the second principal component

where PCA make these lines for datasets to reduce the dimensionalty and there are may be chances of information miss and may not be able to maintian the sparability between clusters.

2) Tha another technique that would be able to maintain the class separability is LDA. The general approach is very similar to PCA, rather than finding the component axes that maximize the variance of our data, we are additionally interested in the axes that maximize the separation between multiple classes.

The goal is to project a dataset onto a lower-dimensional space with good class-separability in order to avoid overfitting and also reduce computational costs.

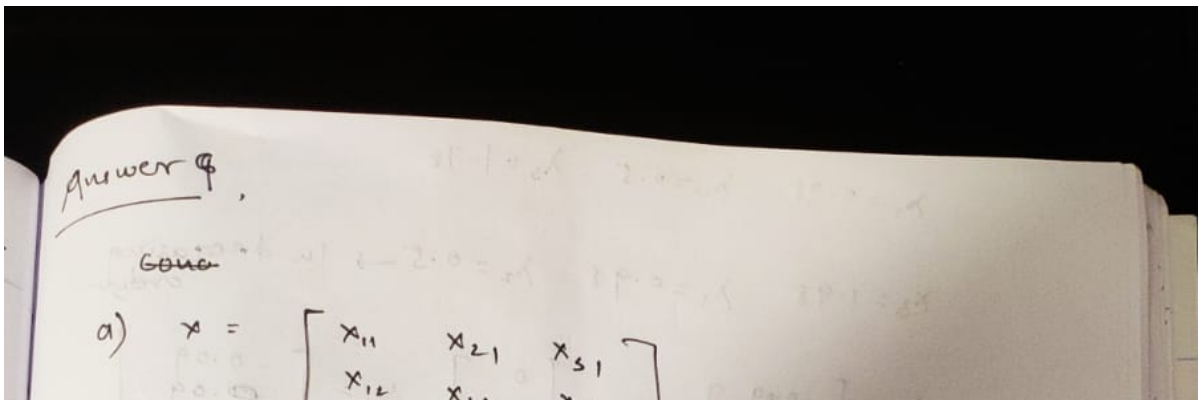
For A, LDA will perform a better dimensionality reduction due to it assumes classes to be compact and separated. For B, LDA can also be used at reducing the dimensions of this dataset due to the classes are clearly separable. For C, PCA will not work due to the points are overlaping in both features and they are not linearly seperable

In [ ]:

Problem 9 (5 points)

Consider the data matrix $\mathbf{X}$ of size $3 \times N$, where N is the number of samples. The covariance matrix $\mathbf{K}$, of size 3×3 has 3 eigenvectors $v_1 = [-0.99, 0.09, 0]^T$, $v_2 = [0, 0, 1]^T$ and $v_3 = [-0.09, -0.99, 0]^T$ with eigenvalues $\lambda_1 = 0.98$, $\lambda_2 = 0.5$ and $\lambda_3 = 1.98$, respectively. Answer the following questions:

1. **What linear transformation would you use to uncorrelate the data $\mathbf{X}$? Provide a numerical solution and justify your answer.**
2. **Use Principal Component Analysis (PCA) to project the 3-dimensional space to a 2-dimensional space. Define the linear transformation (using a numerical answer).**
3. **What is the amount of explained variance of this 2-D projection? Show your work.**
4. **Let $\mathbf{Y}$ be the data (linear) transformation obtained by principal component transform of $\mathbf{X}$ onto a 2-dimensional space. What is resulting covariance matrix of transformed data $\mathbf{Y}$? Use a numerical answer and justify your answer.**



$$\begin{bmatrix} x_{1N} & x_{2N} & x_{3N} \end{bmatrix}_{N\times 3}$$

features = 3

samples = N

$$X^T = \begin{bmatrix} x_{11} & x_{12} & x_{13} & -- & x_{1N} \\ x_{21} & x_{22} & x_{23} & -- & x_{2N} \\ x_{31} & x_{32} & x_{33} & -- & x_{3N} \end{bmatrix}$$

let $U_1, U_2 \& U_3$ be the mean of feature space

$$\bar{x} = \begin{bmatrix} x_{11}-\mu_1 & x_{12}-\mu_1 & x_{13}-\mu_1 & -- & x_{1N}-\mu_1 \\ x_{22}-\mu_2 & x_{22}-\mu_2 & x_{23}-\mu_1 & -- & x_{1N}-\mu_2 \\ x_{31}-\mu_3 & x_{32}-\mu_3 & x_{33}-\mu_3 & -- & x_{3N}-\mu_3 \end{bmatrix}$$

covariance matrix

$$C = \begin{bmatrix} \text{var}(f_1) & \text{cov}(f_1 f_2) & \text{cov}(f_1 f_3) \\ \text{cov}(f_2 f_1) & \text{var}(f_2) & \text{cov}(f_2 f_3) \\ \text{cov}(f_3 f_1) & \text{cov}(f_3 f_2) & \text{var}(f_3) \end{bmatrix}$$

(b) multiply $X_{N\times 3}$ with corelation matrix class

$$X_{N\times 3} = \bar{X}_{N\times 3} * C_{3\times 3}$$

Given eigen value

$$V_1 = \begin{bmatrix} -0.09 \\ 0.09 \\ 0 \end{bmatrix} \quad V_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad V_3 = \begin{bmatrix} -0.09 \\ -0.09 \\ 0 \end{bmatrix}$$

$\lambda_1 = 0.98 \quad \lambda_2 = 0.5 \quad \lambda_3 = 1.78$

$\lambda_3 = 1.98 \quad \lambda_1 = 0.98 \quad \lambda_2 = 0.5 \rightarrow$ in decreasing order.

$$V_2 = \begin{bmatrix} -0.09 \\ -0.09 \\ 0 \end{bmatrix} \quad V_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad V_1 = \begin{bmatrix} -0.09 \\ 0.09 \\ 0 \end{bmatrix}$$

$$K_{SVL} = \begin{bmatrix} -0.09 & -0.09 \\ -0.09 & 0.09 \\ 0 & 0 \end{bmatrix}$$

(2) transform original matrix

$$X_{NOL} = Y_{NVS} = \begin{bmatrix} -0.09 & -0.09 \\ -0.09 & 0.09 \\ 0 & 0 \end{bmatrix}_{3 \times 2}$$

$$Y_3 = \begin{bmatrix} -0.09 & -0.09 \\ -0.09 & -0.09 \\ 0 & 0 \end{bmatrix}$$

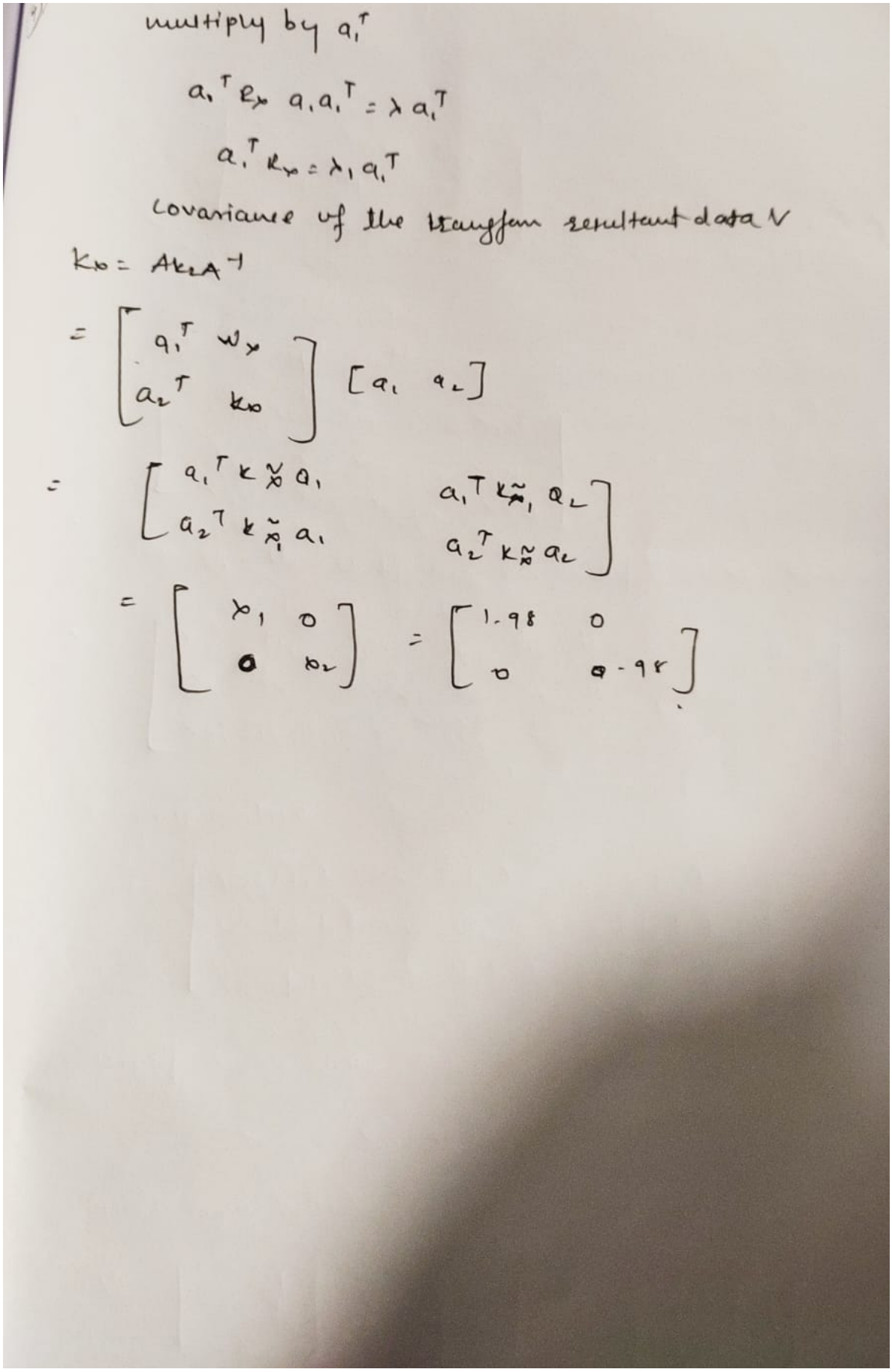
(3) Amount of explained variance

$$P = \frac{\lambda_1 + \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3}$$

$$= \frac{1.98 + 0.98}{1.98 + 0.98 + 0.5}$$

$$= 0.85$$

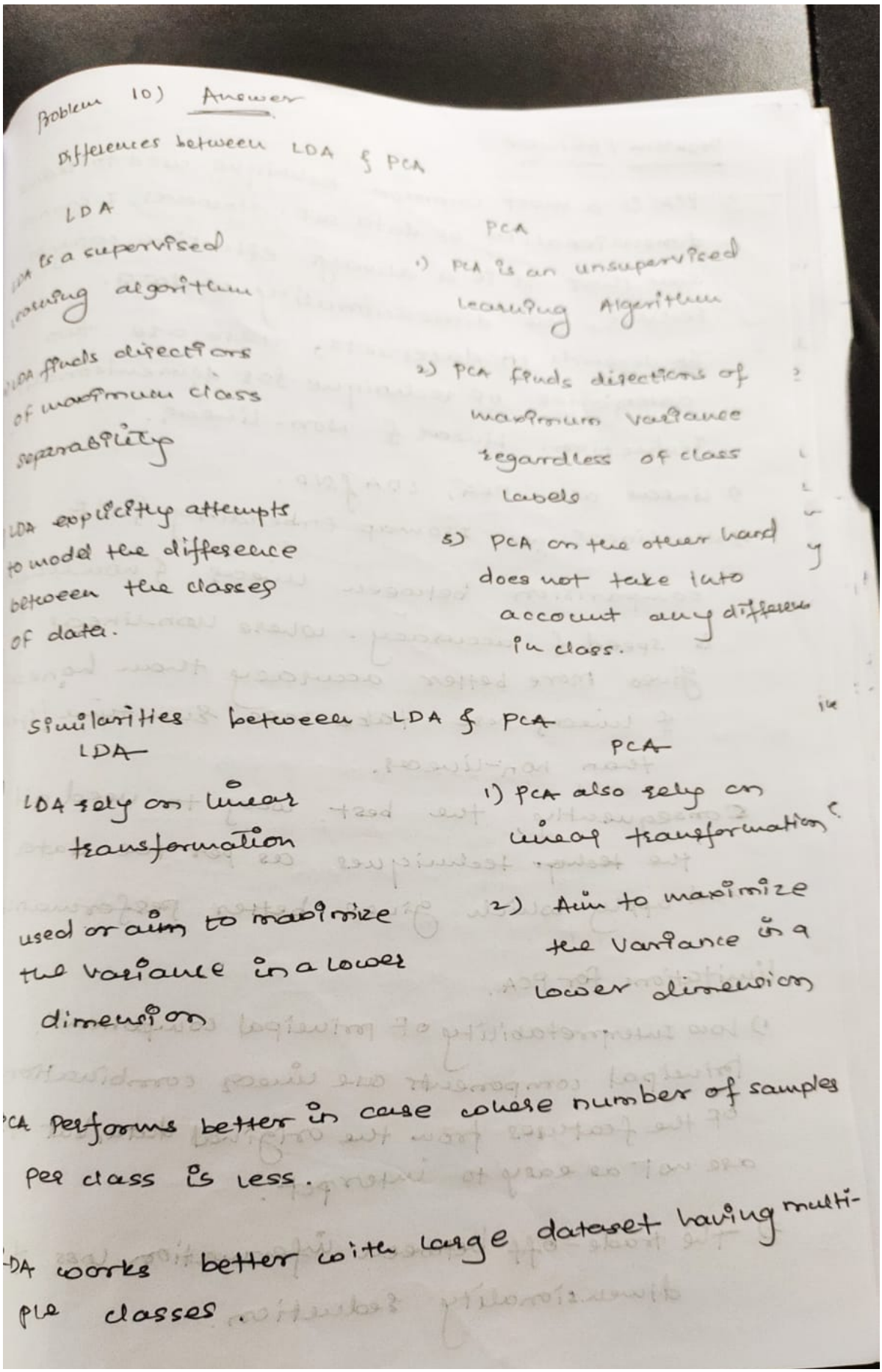
$$= 85.5\%$$


multiply by a_1^T
$a_1^T R_x a_1 a_1^T = \lambda a_1^T$
$a_1^T K_x = \lambda_1 a_1^T$
covariance of the transform resultant data Y
$K_y = A K_x A^{-1}$
$= \begin{bmatrix} a_1^T w_x \\ a_2^T k_y \end{bmatrix} [a_1 \; a_2]$
$= \begin{bmatrix} a_1^T K_x^{\sim} a_1 & a_1^T K_{x_1}^{\sim} a_2 \\ a_2^T K_{x_1}^{\sim} a_1 & a_2^T K_x^{\sim} a_2 \end{bmatrix}$
$= \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} = \begin{bmatrix} 1.98 & 0 \\ 0 & 0.98 \end{bmatrix}$

In [ ]:

Problem 10 (5 points)

What are the differences and similarities between Fisher's Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) as dimensionality reduction approaches? When would you prefer LDA over PCA or vice-versa?



Problem 10) Answer

Differences between LDA & PCA

LDA

1) LDA is a supervised learning algorithm

2) LDA finds directions of maximum class separability

3) LDA explicitly attempts to model the difference between the classes of data.

PCA

1) PCA is an unsupervised learning Algorithm

2) PCA finds directions of maximum variance regardless of class labels

3) PCA on the other hand does not take into account any differences in class.

Similarities between LDA & PCA

LDA

1) LDA rely on linear transformation

2) used or aim to maximize the variance in a lower dimension

PCA

1) PCA also rely on linear transformation.

2) Aim to maximize the variance in a lower dimension.

PCA performs better in case where number of samples per class is less.

LDA works better with large dataset having multiple classes.

In [ ]:

Submit your Solution

Confirm that you've successfully completed the assignment.

Along with the Notebook, include a PDF of the notebook with your solutions.

`add` and `commit` the final version of your work, and `push` your code to your GitHub repository.

Submit the URL of your GitHub Repository as your assignment submission on Canvas.
