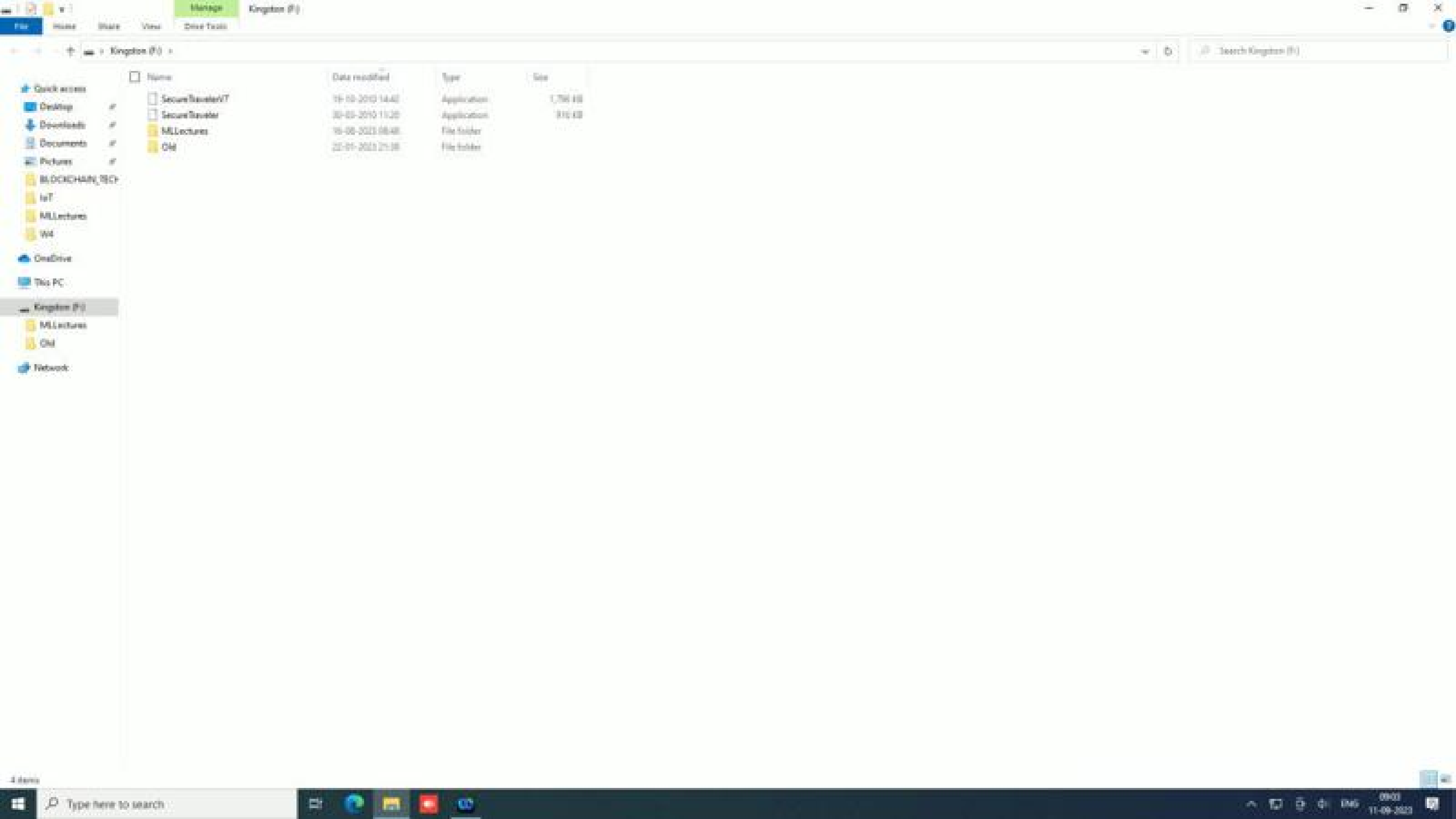
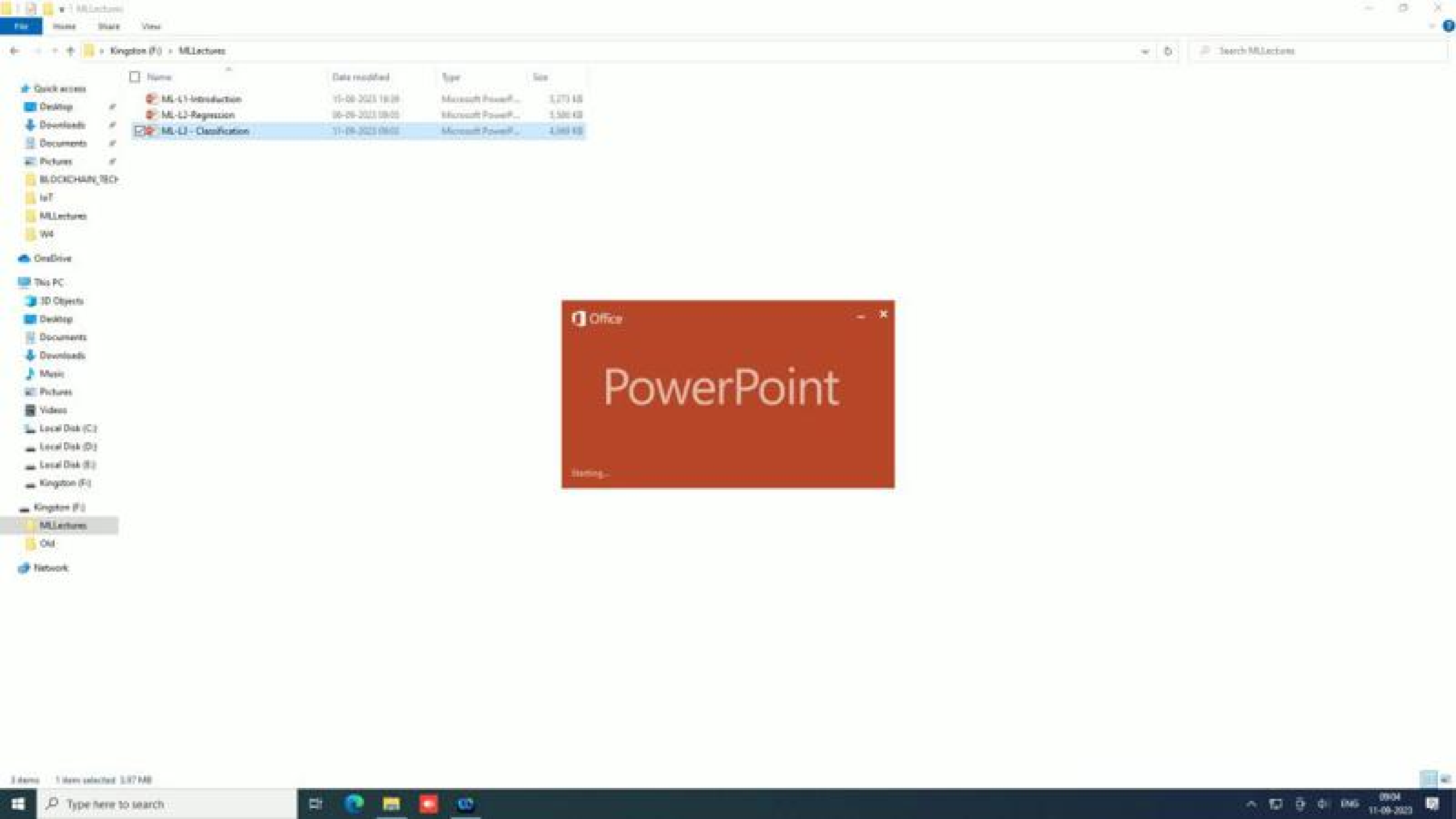
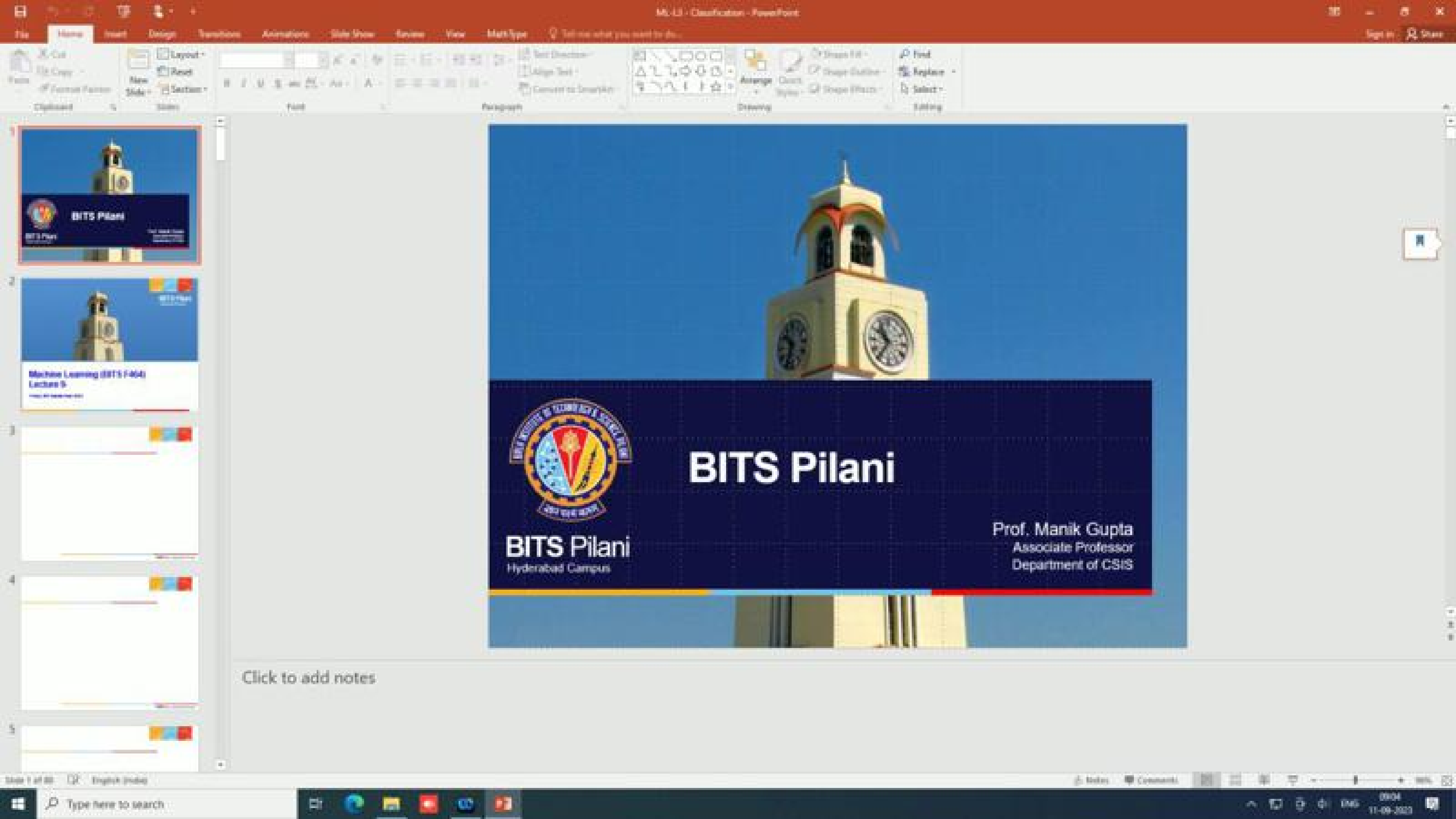


Manage
Kingston (F:)
File
Home
Share
View
Drive Tools
Quick access
Desktop
Downloads
Documents
Pictures
BLOCKCHAIN_TECH
IoT
MLLectures
W4
OneDrive
This PC
Kingston (F:)
MLLectures
Old
Network
Name
Date modified
Type
Size
SecureTravelerV7
Application
SecureTraveler
Application
MLLectures
File folder
Old
File folder
4 items
Type here to search
ENG



File
Home
Share
View
Kingston (F:) > MLLectures
Search MLLectures
Name
Date modified
Type
Size
ML-L1-Introduction
ML-L2-Regression
ML-L3 - Classification
Quick access
Desktop
Downloads
Documents
Pictures
BLOCKCHAIN_TECH
IoT
MLLectures
W4
OneDrive
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Local Disk (D:)
Local Disk (E:)
Kingston (F:)
Kingston (F:)
MLLectures
Old
Network
Office
PowerPoint
Starting...
Type here to search
ENG


BITS Pilani
BITS Pilani
Hyderabad Campus
Prof. Manik Gupta
Associate Professor
Department of CSIS
Click to add notes

Agenda

Linear Models for Classification

- Discriminant Functions
- Least square classification
- Fishers Linear Discriminant
- Perceptron

Agenda

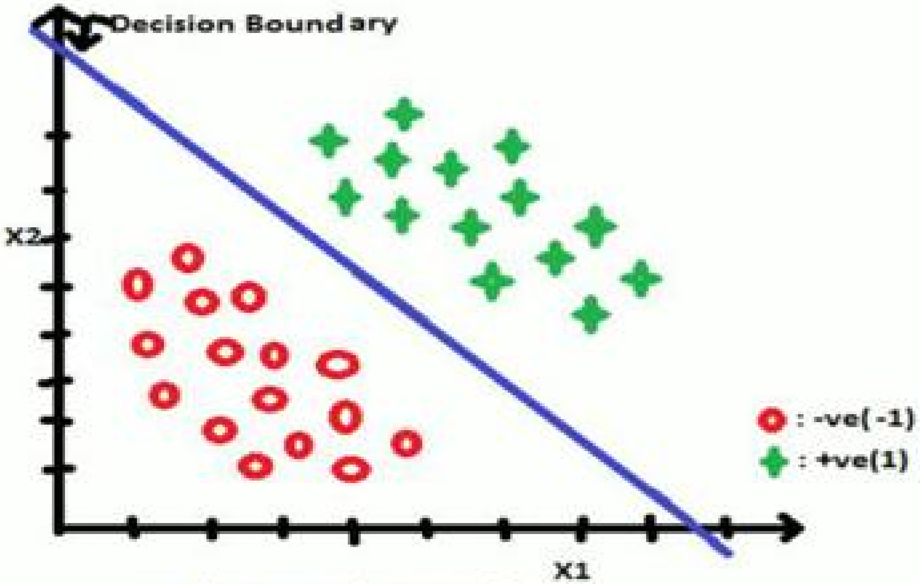
Linear Models for Classification

- Discriminant Functions
- Least square classification
- Fishers Linear Discriminant
- Perceptron

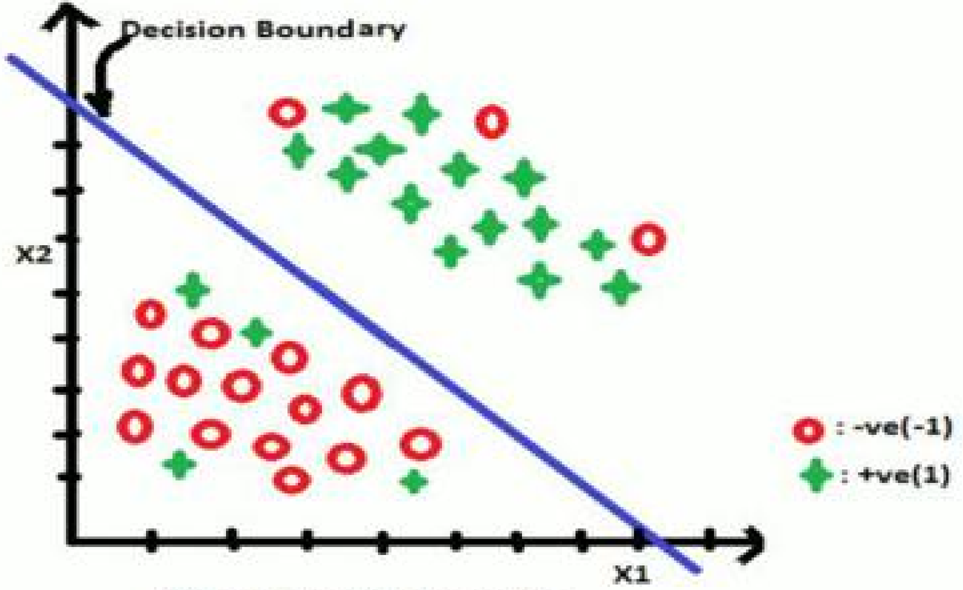
Introduction

- The goal in classification is to take an input vector $\boldsymbol{x}$ and to assign it to one of K discrete classes C_k where $k = 1, \ldots, K$.
- In the most common scenario, the classes are taken to be disjoint, so that each input is assigned to one and only one class.
- The input space is thereby divided into *decision regions* whose boundaries are called *decision boundaries* or *decision surfaces*.

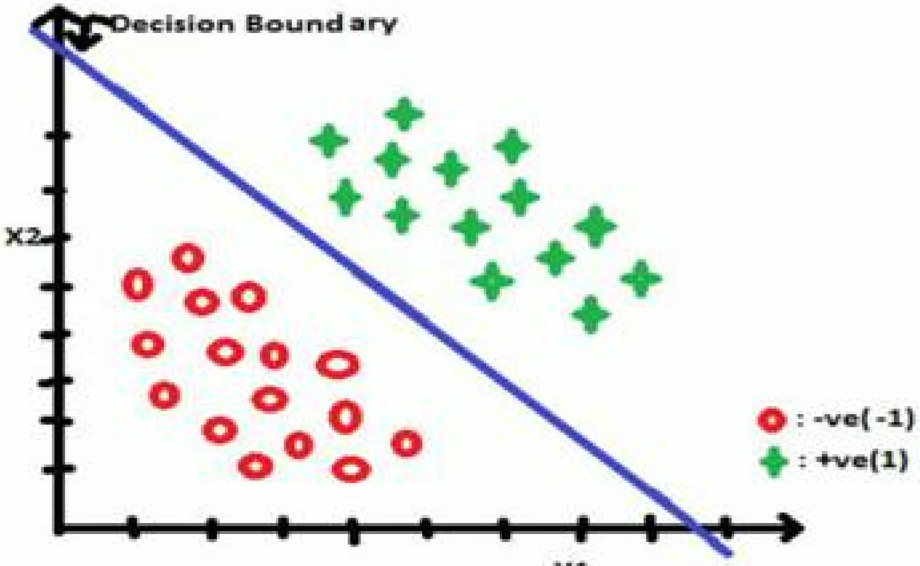
- We will consider linear models for classification, by which we mean that the decision surfaces are linear functions of the input vector $\boldsymbol{x}$ and hence are defined by $(D-1)$-dimensional hyperplanes within the D-dimensional input space.
- Datasets whose classes can be separated exactly by linear decision surfaces are said to be *linearly separable*.



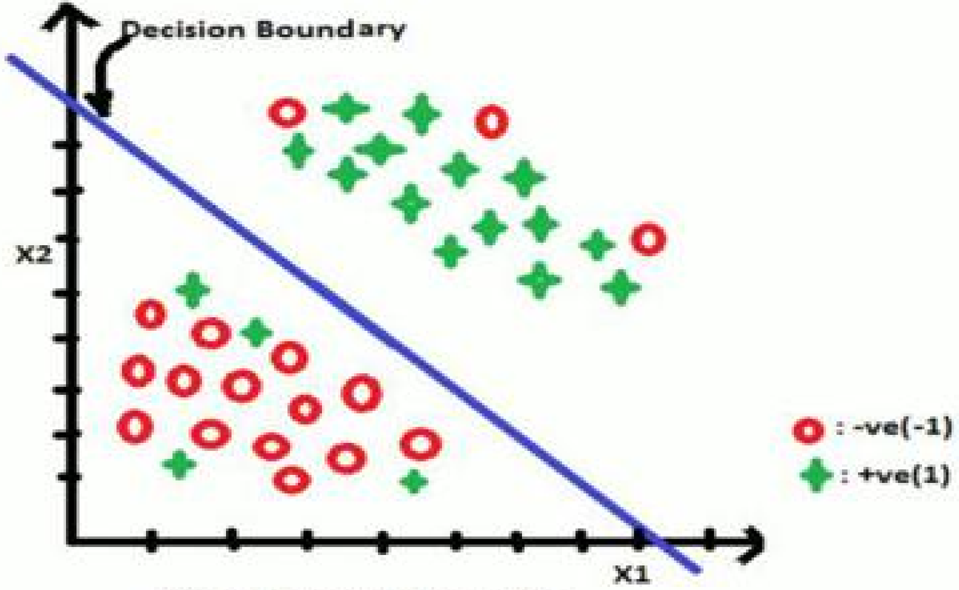
Perfect Linearly separable



Almost Linearly Separable



Perfect Linearly separable



Almost Linearly Separable

- For regression problems, the target variable $\boldsymbol{t}$ was simply the vector of real numbers whose values we wish to predict.
- In the case of classification, there are various ways of using target values to represent class labels.
- For probabilistic models, the most convenient, in the case of two-class problems –
  - Binary representation in which there is a single target variable $t \in \{0, 1\}$ such that $t = 1$ represents class C_1 and $t = 0$ represents class C_2.
  - We can interpret the value of t as the probability that the class is C_1, with the values of probability taking only the extreme values of 0 and 1.

- For $K > 2$ classes, it is convenient to use a $1 - of - K$ coding scheme in which $\boldsymbol{t}$ is a vector of length K such that if the class is C_j , then all elements t_k of $\boldsymbol{t}$ are zero except element tj , which takes the value 1.
- For instance, if we have $K = 5$ classes, then a pattern from class 2 would be given the target vector

$$\mathbf{t} = (0, 1, 0, 0, 0)^{\mathrm{T}}.$$

Generalized linear models

- In the linear regression models considered, the model prediction $y(\boldsymbol{x}, \boldsymbol{w})$ was given by a linear function of the parameters $\boldsymbol{w}$.
- In the simplest case, the model is also linear in the input variables and therefore takes the form $y(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0$, so that y is a real number.

- For classification problems, however, we wish to predict discrete class labels, or more generally posterior probabilities that lie in the range (0, 1).

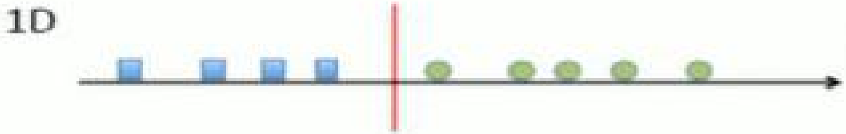
- To achieve this, we consider a generalization of this model in which we transform the linear function of $\boldsymbol{w}$ using a nonlinear function

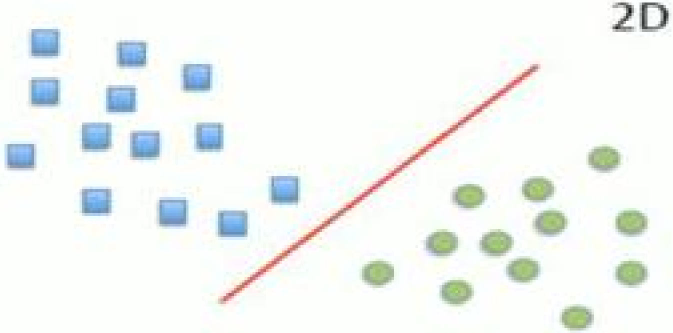
$f(\cdot)$ so that

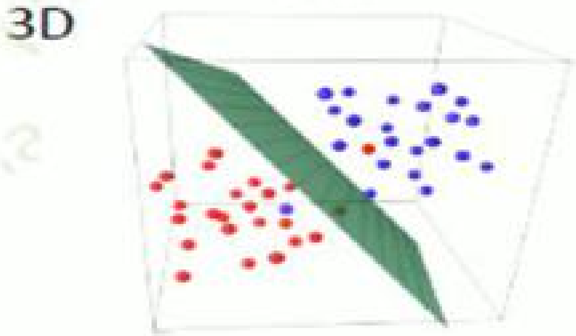
$$y(\mathbf{x}) = f\left(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0\right).$$

- In the machine learning literature, $f(\cdot)$ is known as an *activation function*, whereas its inverse is called a *link function* in the statistics literature.
- The decision surfaces correspond to $y(\boldsymbol{x}) = constant$, so that $\boldsymbol{w}^T\boldsymbol{x} + w_0 = constant$ and hence the decision surfaces are linear functions of $\boldsymbol{x}$, even if the function $f(\cdot)$ is nonlinear.
- For this reason, this class of models are called *generalized linear models*

Constant decision surfaces







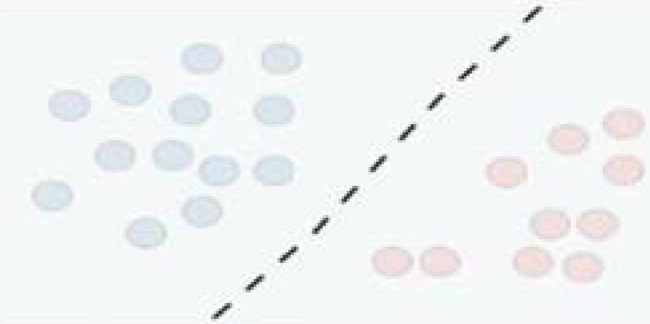
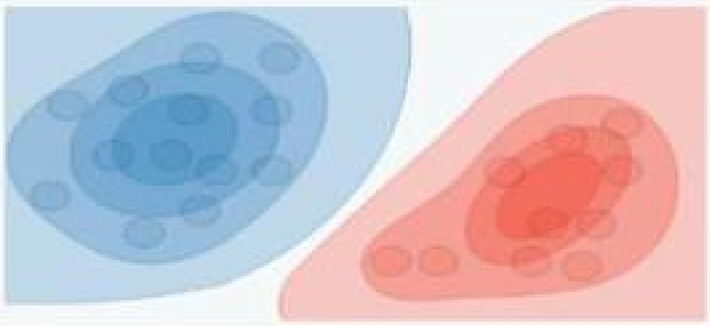
$$w_0 + \mathbf{w}^T \mathbf{x} = 0$$

All are linear boundaries

- Three distinct approaches to the classification problem.
  - The simplest involves constructing a **discriminant function** that directly assigns each vector $\boldsymbol{x}$ to a specific class.
  - A more powerful approach, however, models the conditional probability distribution $p(C_k|\boldsymbol{x})$ in an inference stage, and then subsequently uses this distribution to make optimal decisions.

- There are two different approaches to determining the conditional probabilities $p(C_k|\boldsymbol{x})$.

- There are two different approaches to determining the conditional probabilities $p(C_k|\boldsymbol{x})$.
  - One technique is to model them directly, for example by representing them as parametric models and then optimizing the parameters using a training set
  - Alternatively, we can adopt a generative approach in which we model the **class-conditional densities** given by $p(x|C_k)$, together with the **prior probabilities** $p(C_k)$ for the classes, and then we compute the required posterior probabilities using Bayes' theorem.

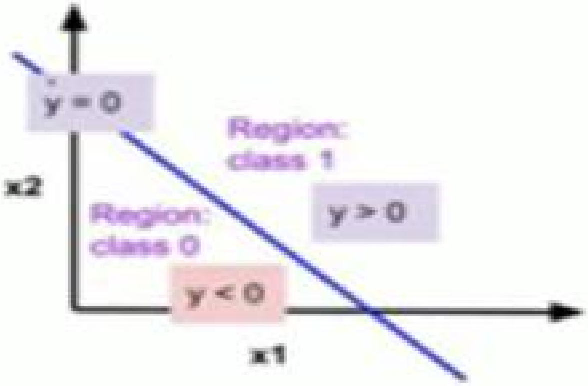
	Discriminative model	Generative model
Goal	Directly estimate $P(y\|x)$	Estimate $P(x\|y)$ to then deduce $P(y\|x)$
What's learned	Decision boundary	Probability distributions of the data
Illustration		
Examples	Regressions, SVMs	GDA, Naive Bayes

Discriminant Functions

- A discriminant is a function that takes an input vector $\boldsymbol{x}$ and assigns it to one of K classes, denoted C_k.

Discriminant functions learn direct mapping between feature vector $\mathbf{x}$ and label y.

- An input vector $\boldsymbol{x}$ is assigned to class C_1 if $y(\boldsymbol{x}) >= 0$ and to class C_2 otherwise.
- The corresponding decision boundary is therefore defined by the relation $y(\boldsymbol{x}) = 0$, which corresponds to a $(D-1)$-dimensional hyperplane within the D $-$dimensional input space.



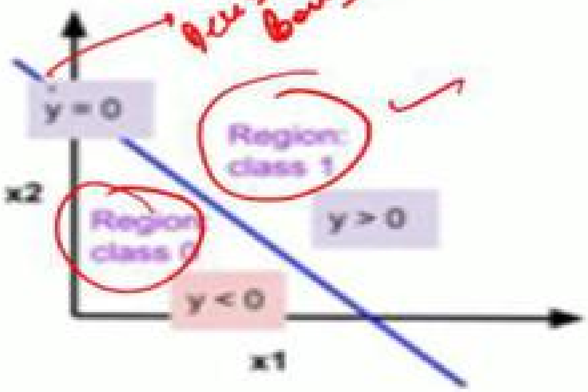
- The simplest representation of a linear discriminant function is obtained by taking a linear function of the input vector so that

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$$

where w is the weight vector and w_0 is a bias.

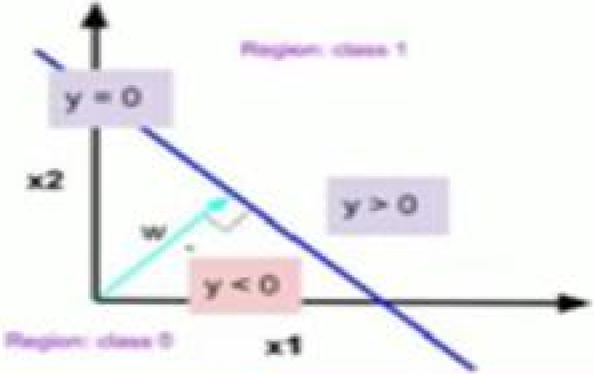
- An input vector $\boldsymbol{x}$ is assigned to class C_1 if $y(\boldsymbol{x}) >= 0$ and to class C_2 otherwise.
- The corresponding decision boundary is therefore defined by the relation $y(\boldsymbol{x}) = 0$, which corresponds to a $(D-1)$-dimensional hyperplane within the D $-$dimensional input space.

decision boundary



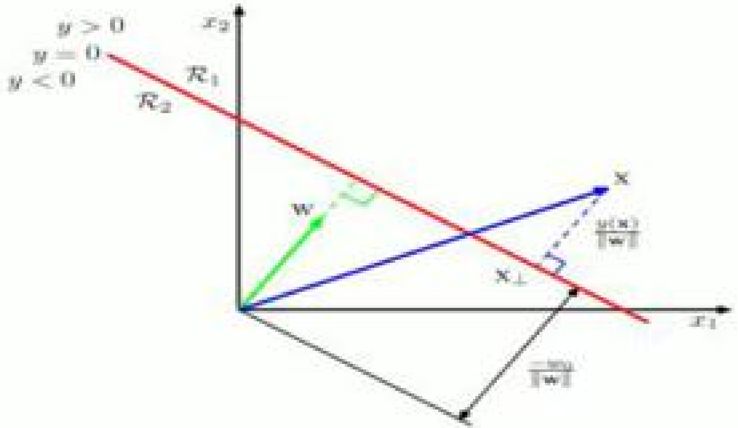
- Consider two points $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$ both of which lie on the decision surface.
- Because $y(\boldsymbol{x}_A) = y(\boldsymbol{x}_B) = 0$, we have $\boldsymbol{w}^T(\boldsymbol{x}_A - \boldsymbol{x}_B) = 0$ and hence the vector $\boldsymbol{w}$ is orthogonal to every vector lying within the decision surface, and so $\boldsymbol{w}$ **determines the orientation of the decision surface.**
- Similarly, if $\boldsymbol{x}$ is a point on the decision surface, then $y(\boldsymbol{x}) = 0$, and so the normal distance from the origin to the decision surface is given by

$$\frac{\mathbf{w}^{\mathrm{T}}\mathbf{x}}{\|\mathbf{w}\|} = -\frac{w_0}{\|\mathbf{w}\|}.$$



- **Value of $y(x)$ gives a signed measure of the perpendicular distance r of the point x from the decision surface.**
- Consider an arbitrary point x and let $x_{\perp}$ be its orthogonal projection onto the decision surface, so that

$$\mathbf{x} = \mathbf{x}_{\perp} + r\frac{\mathbf{w}}{\|\mathbf{w}\|}.$$

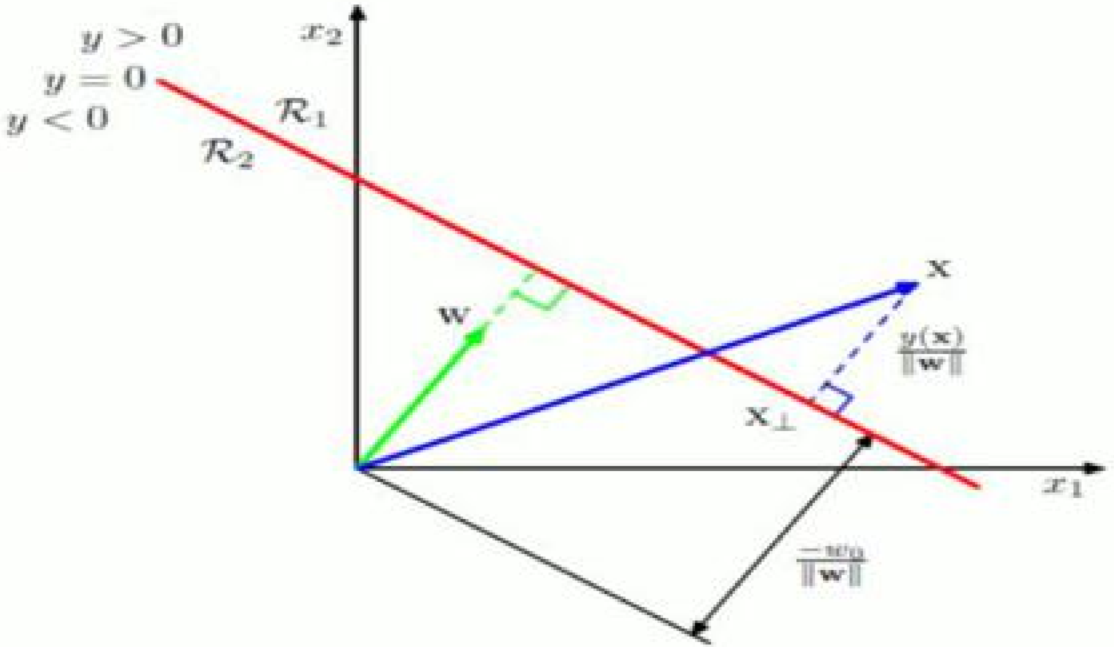


- Multiplying both sides of this result by w^T and adding w_0, and making use of $y(x) = w^T x + w_0$ and $y(x_{\perp}) = w^T x_{\perp} + w_0 = 0$, we have

$$r = \frac{y(\mathbf{x})}{\|\mathbf{w}\|}.$$

Geometry of discriminant function in 2D

- w_0 determines the location of the decision surface.
- $\mathbf{w}$ determines the orientation of the decision surface.
- y gives signed measure of perpendicular distance of the point $\mathbf{x}$ from the decision surface.
- Decision surface divides feature space into two regions.

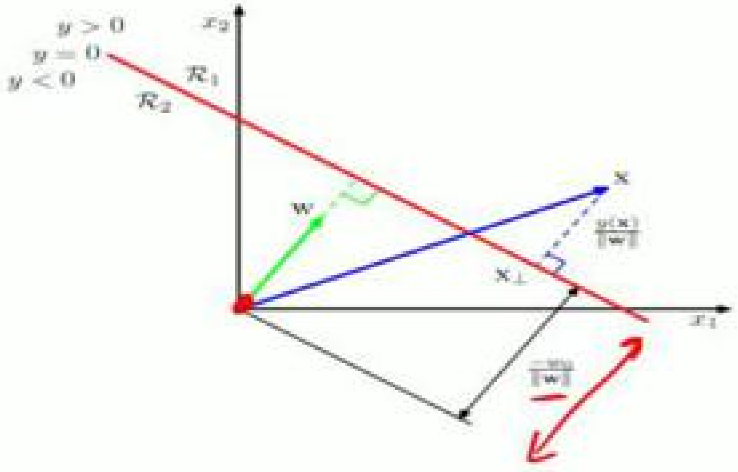


- **Value of $y(x)$ gives a signed measure of the perpendicular distance r of the point x from the decision surface.**
- Consider an arbitrary point $\boldsymbol{x}$ and let $\boldsymbol{x}_{\perp}$ be its orthogonal projection onto the decision surface, so that

$$\mathbf{x} = \mathbf{x}_{\perp} + r\frac{\mathbf{w}}{\|\mathbf{w}\|}.$$

- Multiplying both sides of this result by $\boldsymbol{w}^T$ and adding w_0, and making use of $y(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0$ and $y(\boldsymbol{x}_{\perp}) = \boldsymbol{w}^T\boldsymbol{x}_{\perp} + w_0 = 0$, we have

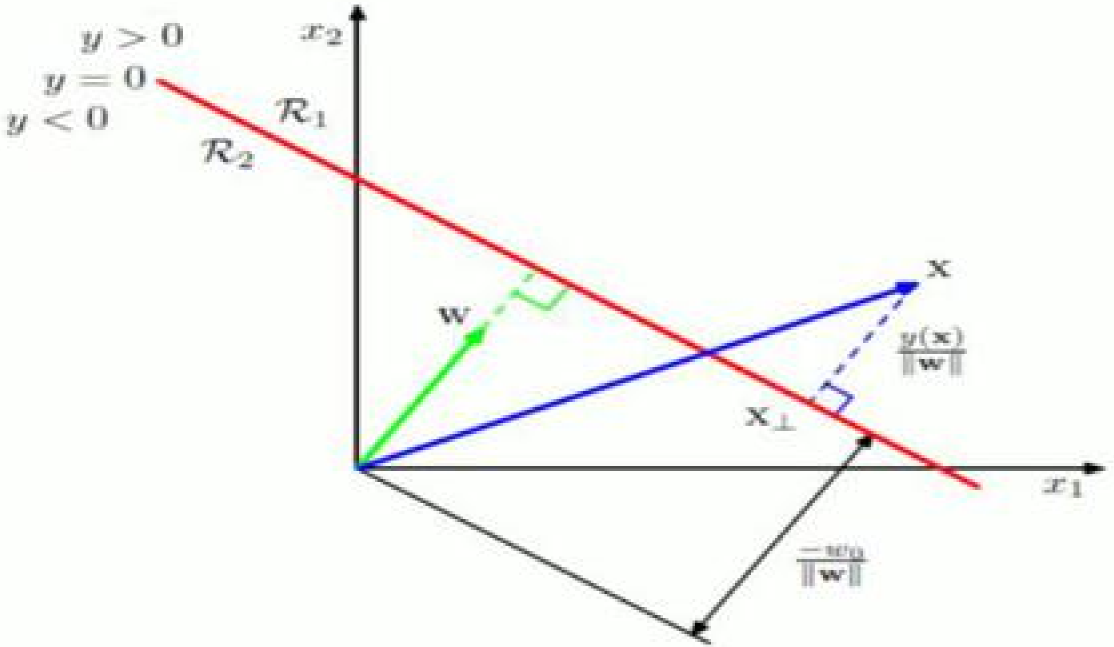
$$r = \frac{y(\mathbf{x})}{\|\mathbf{w}\|}.$$



$-\frac{w_0}{\|w\|}$

Geometry of discriminant function in 2D

- w_0 determines the location of the decision surface.
- $\mathbf{w}$ determines the orientation of the decision surface.
- y gives signed measure of perpendicular distance of the point $\mathbf{x}$ from the decision surface.
- Decision surface divides feature space into two regions.



Agenda

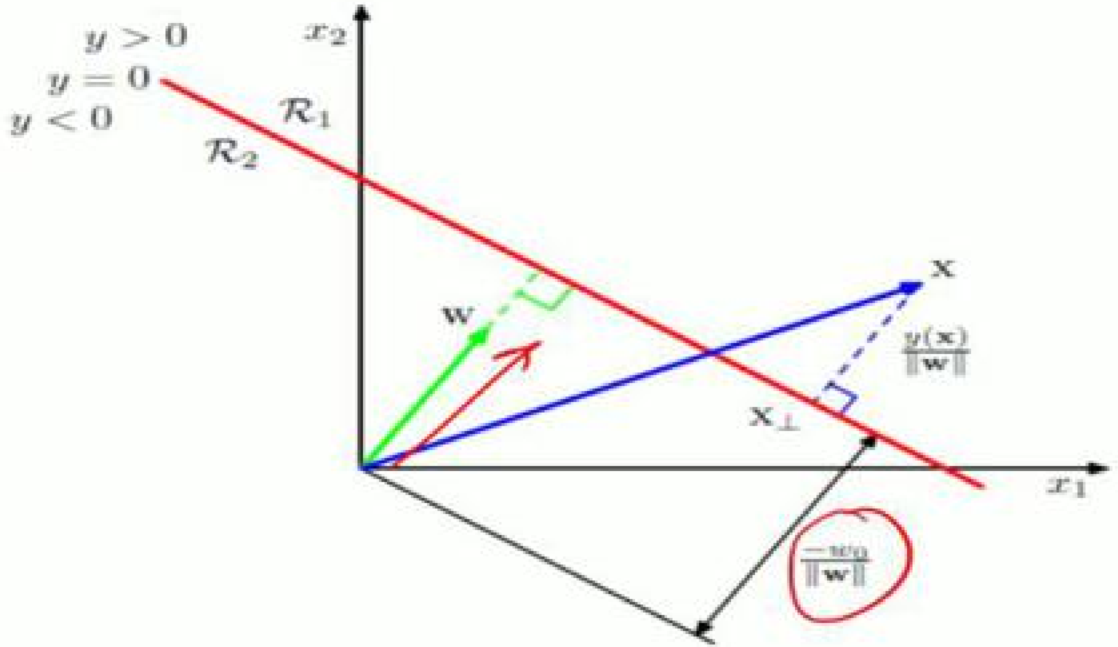
Linear Models for Classification

- Discriminant Functions
- Least square classification
- Fishers Linear Discriminant
- Perceptron

Click to add notes

Geometry of discriminant function in 2D

- w_0 determines the location of the decision surface.
- $\mathbf{w}$ determines the orientation of the decision surface.
- y gives signed measure of perpendicular distance of the point $\mathbf{x}$ from the decision surface.
- Decision surface divides feature space into two regions.



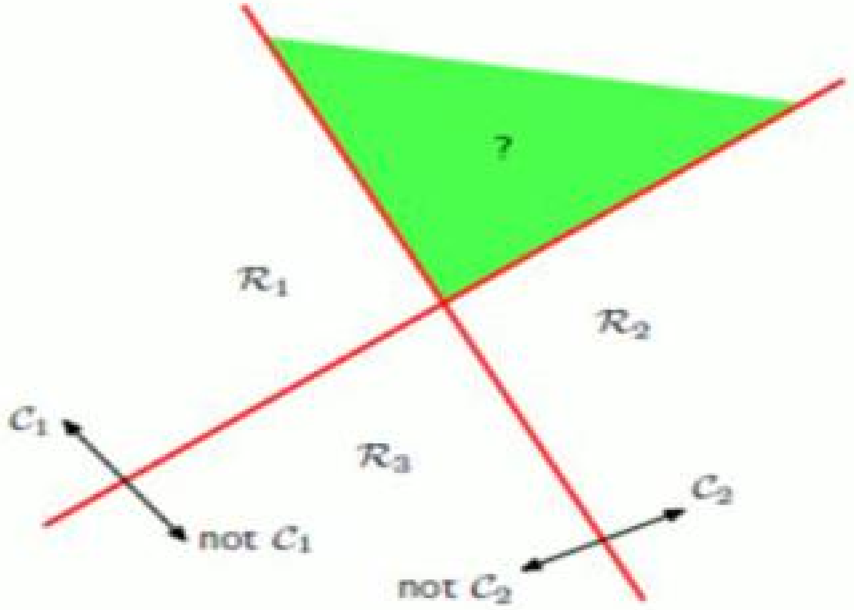
Multiple Classes

- Assuming the number of classes to be K>2, the discriminant functions can be build in two ways by combining a number of two-class discriminant functions

  - **One vs rest** – build K-1 discriminant functions. Each discriminant function solves a two class classification problem.

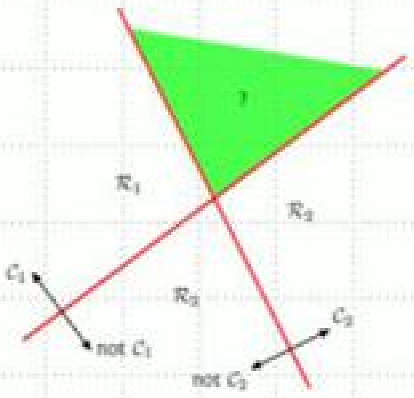
  - **One vs one** – one discriminant function per pair of classes. Each point is then classified according to a majority vote amongst the discriminant functions.

Issues with one-vs-rest



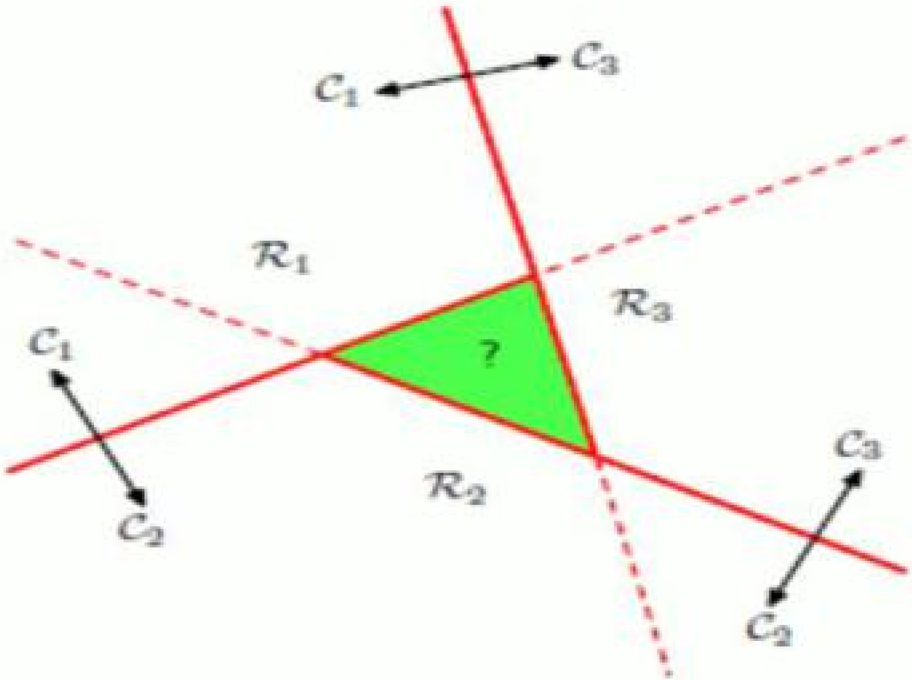
Issues with one-vs-rest

Click to add text

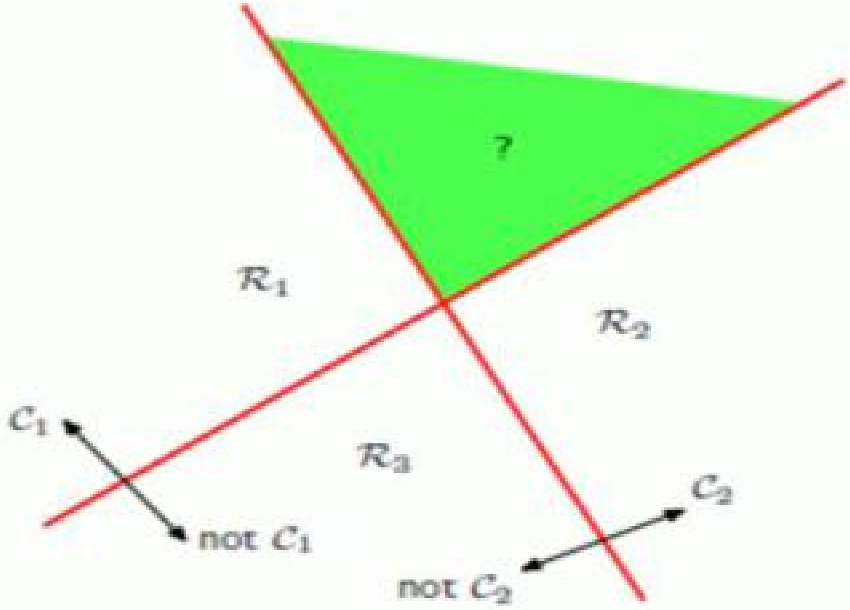


Click to add notes

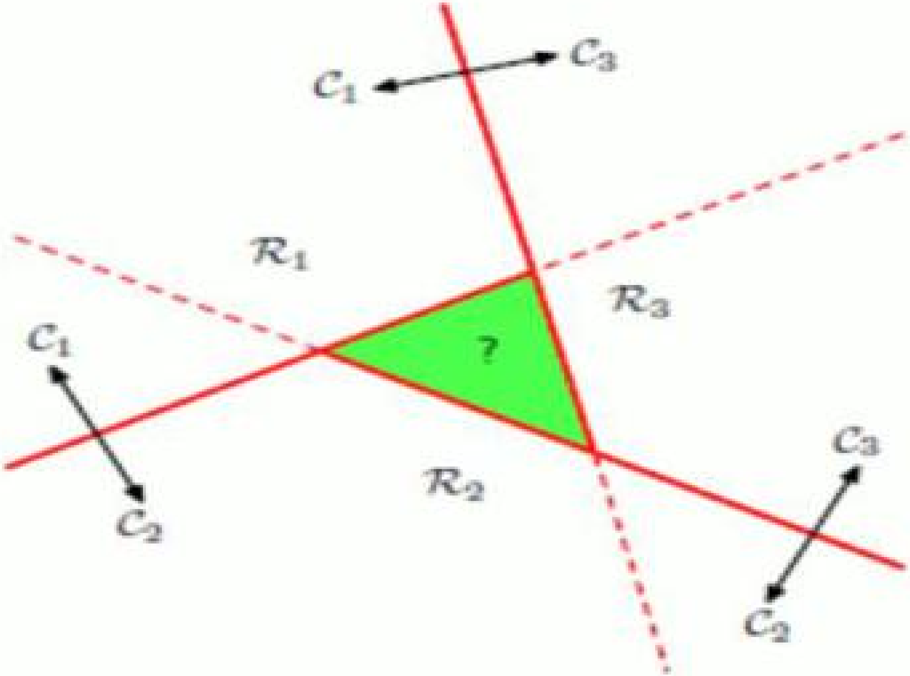
Issues with one-vs-one



Issues with one-vs-rest



Issues with one-vs-one



- Considering a single K-class discriminant comprising K linear functions of the form

$$y_k(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0}$$

and then assigning a point $\boldsymbol{x}$ to class C_k if $y_k(\boldsymbol{x}) > y_j(\boldsymbol{x})$ for all $j \neq k$.

- The decision boundary between class C_k and class C_j is therefore given by $y_k(\boldsymbol{x}) = y_j(\boldsymbol{x})$ and hence corresponds to a $(D-1)$-dimensional hyperplane defined by

$$(\mathbf{w}_k - \mathbf{w}_j)^{\mathrm{T}}\mathbf{x} + (w_{k0} - w_{j0}) = 0.$$

- Three approaches to learning the parameters of linear discriminant functions, based on
  - Least Squares
  - Fisher's Linear Discriminant
  - Perceptron Algorithm

Least square classification

- Consider a classification problem with K classes, with a 1-of-K binary coding scheme for the target vector t.
- Each class C_k has its own linear model.
- Each class C_k is described by its own linear model so that

$$y_k(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0}$$

- This can be rewritten as

$$\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^{\mathrm{T}}\widetilde{\mathbf{x}}$$

where , $\widetilde{\boldsymbol{W}}$ is a matrix whose k^{th} column comprises the D + 1-dimensional vector $\widetilde{\boldsymbol{w}}_k = (w_{k0}, \boldsymbol{w}_{\boldsymbol{k}}^{\boldsymbol{T}})^{\mathrm{T}}$ and $\widetilde{\boldsymbol{x}}$ is the corresponding augmented input vector $(1, \boldsymbol{x}^T)^T$ with a dummy input $x_0 = 1$.

$$\widetilde{W} = \begin{bmatrix} w_1^0 & \cdots & w_K^0 \\ \vdots & \vdots & \vdots \\ w_1^D & \cdots & w_K^D \end{bmatrix}$$

- A new input $\boldsymbol{x}$ is then assigned to the class for which the output $y_k = \widetilde{w}_k^T \widetilde{x}$ is largest.

Least square classification

- Consider a classification problem with K classes, with a 1-of-K binary coding scheme for the target vector t. (5)
- Each class C_k has its own linear model.
- Each class C_k is described by its own linear model so that

$$y_k(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0}$$

- This can be rewritten as

$$\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^{\mathrm{T}}\widetilde{\mathbf{x}}$$

where , $\widetilde{\boldsymbol{W}}$ is a matrix whose k^{th} column comprises the D + 1-dimensional vector $\widetilde{\boldsymbol{w}}_k = (w_{k0}, \boldsymbol{w}_{\boldsymbol{k}}^{T})^{\mathrm{T}}$ and $\widetilde{\boldsymbol{x}}$ is the corresponding augmented input vector $(1, \boldsymbol{x}^T)^T$ with a dummy input $x_0 = 1$.

$$\widetilde{W} = \begin{bmatrix} w_1^0 & \cdots & w_K^0 \\ \vdots & \vdots & \vdots \\ w_1^D & \cdots & w_K^D \end{bmatrix}$$

- A new input $\boldsymbol{x}$ is then assigned to the class for which the output $y_k = \widetilde{w}_k^T \widetilde{x}$ is largest.

Least square classification

- Consider a classification problem with K classes, with a 1-of-K binary coding scheme for the target vector t. (y)
- Each class C_k has its own linear model.
- Each class C_k is described by its own linear model so that

$$y_k(\mathbf{x}) = \mathbf{w}_k^{\mathbf{T}}\mathbf{x} + w_{k0}$$

- This can be rewritten as

$$\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^{\mathrm{T}}\widetilde{\mathbf{x}}$$

where , $\widetilde{\boldsymbol{W}}$ is a matrix whose k^{th} column comprises the D + 1-dimensional vector $\widetilde{\boldsymbol{w}}_k = (w_{k0}, \boldsymbol{w}_{\boldsymbol{k}}^{\boldsymbol{T}})^{\mathrm{T}}$ and $\widetilde{\boldsymbol{x}}$ is the corresponding augmented input vector $(1, \boldsymbol{x}^T)^T$ with a dummy input $x_0 = 1$.

$$\widetilde{W} = \begin{bmatrix} w_1^0 & \cdots & w_K^0 \\ \vdots & \vdots & \vdots \\ w_1^D & \cdots & w_K^D \end{bmatrix}$$

- A new input $\boldsymbol{x}$ is then assigned to the class for which the output $y_k = \widetilde{w}_k^T \widetilde{x}$ is largest.

- We now determine the parameter matrix , $\widetilde{\boldsymbol{W}}$ by minimizing a sum-of-squares error function
- Consider a training data set $\{\boldsymbol{x}_n, \boldsymbol{t}_n\}$ where $n = 1, \ldots, N$, and define a matrix $\boldsymbol{T}$ whose n^{th} row is the vector $\boldsymbol{t}_{\boldsymbol{n}}^{\boldsymbol{T}}$, together with a matrix $\boldsymbol{X}$ whose n^{th} row is $\boldsymbol{x}_{\boldsymbol{n}}^{\boldsymbol{T}}$

$$T = \begin{bmatrix} t_1^1 & \dots & t_1^K \\ \vdots & \vdots & \vdots \\ t_N^1 & \dots & t_N^K \end{bmatrix}, \tilde{X} = \begin{bmatrix} x_1^0 & \dots & x_1^D \\ \vdots & \vdots & \vdots \\ x_N^0 & \dots & x_N^D \end{bmatrix}$$

- The sum-of-squares error function can then be written as

$$E_D(\widetilde{\mathbf{W}}) = \frac{1}{2}\mathrm{Tr}\left\{(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})^{\mathrm{T}}(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})\right\}.$$

Least square classification

- Consider a classification problem with K classes, with a 1-of-K binary coding scheme for the target vector t. (y)
- Each class C_k has its own linear model.
- Each class C_k is described by its own linear model so that

$$y_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x} + w_{k0}$$

- This can be rewritten as

$$\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^{\mathrm{T}}\widetilde{\mathbf{x}}$$

where , $\widetilde{\boldsymbol{W}}$ is a matrix whose k^{th} column comprises the D + 1-dimensional vector $\widetilde{\boldsymbol{w}}_k = (w_{k0}, \boldsymbol{w}_k^T)^{\mathrm{T}}$ and
$\widetilde{\boldsymbol{x}}$ is the corresponding augmented input vector $(1, \boldsymbol{x}^T)^T$ with a dummy input $x_0 = 1$.

$$\widetilde{W} = \begin{bmatrix} w_1^0 & \cdots & w_K^0 \\ \vdots & \vdots & \vdots \\ w_1^D & \cdots & w_K^D \end{bmatrix}$$

- A new input $\boldsymbol{x}$ is then assigned to the class for which the output $y_k = \widetilde{w}_k^T \tilde{x}$ is largest.

- We now determine the parameter matrix , $\widetilde{W}$ by minimizing a sum-of-squares error function
- Consider a training data set $\{\boldsymbol{x}_n, \boldsymbol{t}_n\}$ where $n = 1, \ldots, N$, and define a matrix $\boldsymbol{T}$ whose n^{th} row is the vector $\boldsymbol{t}_n^T$, together with a matrix $\boldsymbol{X}$ whose n^{th} row is $\boldsymbol{x}_n^T$

$$T = \begin{bmatrix} t_1^1 & \cdots & t_1^K \\ \vdots & \vdots & \vdots \\ t_N^1 & \cdots & t_N^K \end{bmatrix}, \tilde{X} = \begin{bmatrix} x_1^0 & \cdots & x_1^D \\ \vdots & \vdots & \vdots \\ x_N^0 & \cdots & x_N^D \end{bmatrix}$$

3 4

$$T = \begin{array}{cc} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \begin{array}{l} \to c1 \\ \to c2 \\ \to c3 \\ \to c4 \end{array} \end{array}$$

- The sum-of-squares error function can then be written as

$$E_D(\widetilde{\mathbf{W}}) = \frac{1}{2}\mathrm{Tr}\left\{(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})^{\mathrm{T}}(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})\right\}.$$

- This can be rewritten as

$$\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^{\mathrm{T}}\widetilde{\mathbf{x}}$$

where , $\widetilde{\boldsymbol{W}}$ is a matrix whose k^{th} column comprises the D + 1-dimensional vector $\widetilde{\boldsymbol{w}}_k = (w_{k0}, \boldsymbol{w}_{\boldsymbol{k}}^{\boldsymbol{T}})^{\mathrm{T}}$ and $\widetilde{\boldsymbol{x}}$ is the corresponding augmented input vector $(1, \boldsymbol{x}^T)^T$ with a dummy input $x_0 = 1$.

$$\widetilde{W} = \begin{bmatrix} w_1^0 & \cdots & w_K^0 \\ \vdots & \vdots & \vdots \\ w_1^D & \cdots & w_K^D \end{bmatrix}$$

- A new input $\boldsymbol{x}$ is then assigned to the class for which the output $y_k = \widetilde{w}_k^T \tilde{x}$ is largest.

$$\mathbf{y}(\mathbf{x}) = \tilde{\mathbf{W}}^T \tilde{\mathbf{x}}$$

(D+1) × K matrix whose kth column comprises of D+1 dimensional vector:

$$\tilde{\mathbf{w}}_k = (w_{k0}, \mathbf{w}_k^T)^T.$$

corresponding augmented input vector:

$$\tilde{\mathbf{x}} = (1, \mathbf{x}^T)^T.$$

$$\tilde{\mathbf{W}} = (\tilde{\mathbf{X}}^T \tilde{\mathbf{X}})^{-1} \tilde{\mathbf{X}}^T \mathbf{T}$$

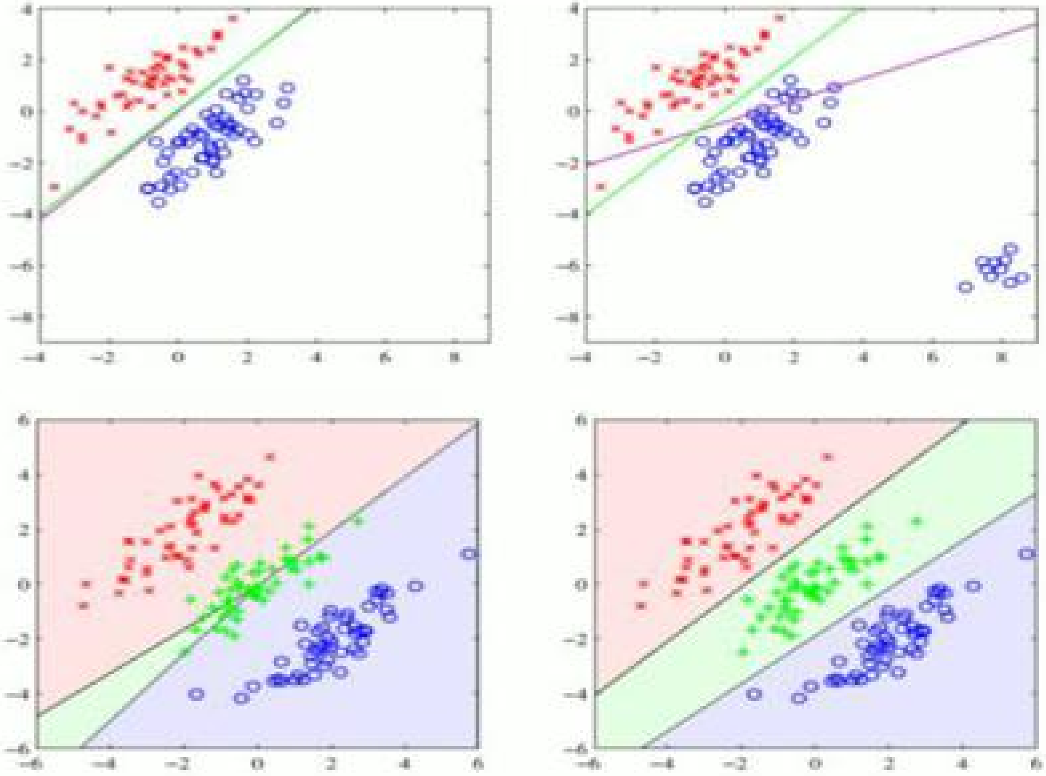
Optimal weights

N × (D+1) input matrix whose n^{th} row is $\tilde{\mathbf{x}}_n^T$.

N × K target matrix whose n^{th} row is $\mathbf{t}_n^T$.

Issues with LSC

- Least-squares solutions lack robustness to outliers and can give poor results.

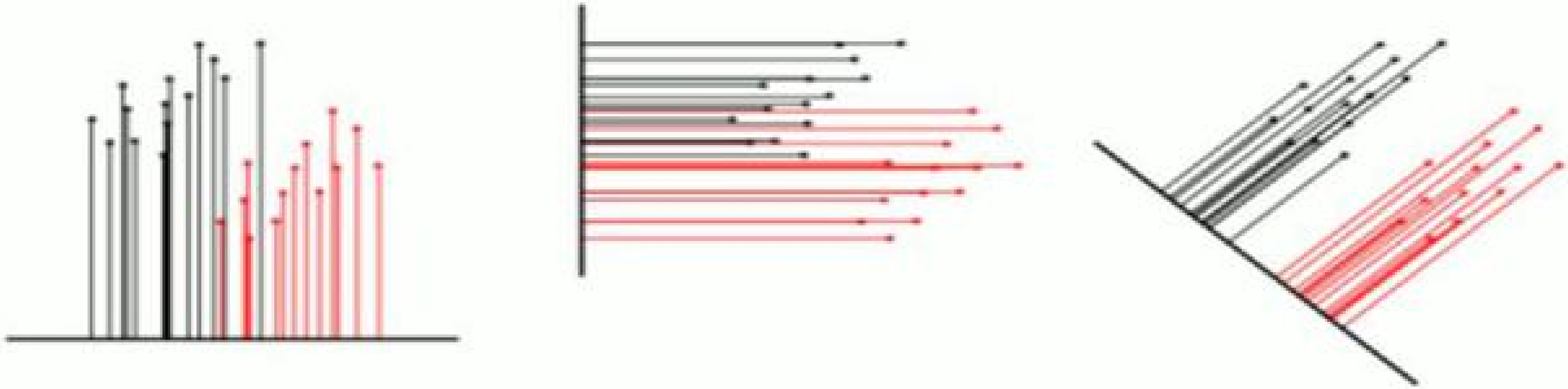


Fisher's linear discriminant

- One way to view a linear classification model is in terms of dimensionality reduction.
- Suppose we take a D-dim input vector and project it down to one dimension using $y = \boldsymbol{w}^T \boldsymbol{x}$

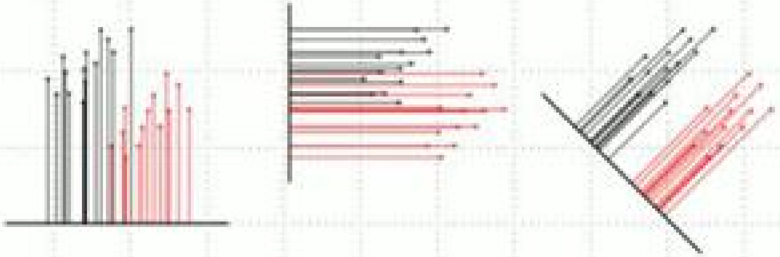
- Main Idea: Find the projection that maximizes the class separation.

Projection of data from two classes onto various lines



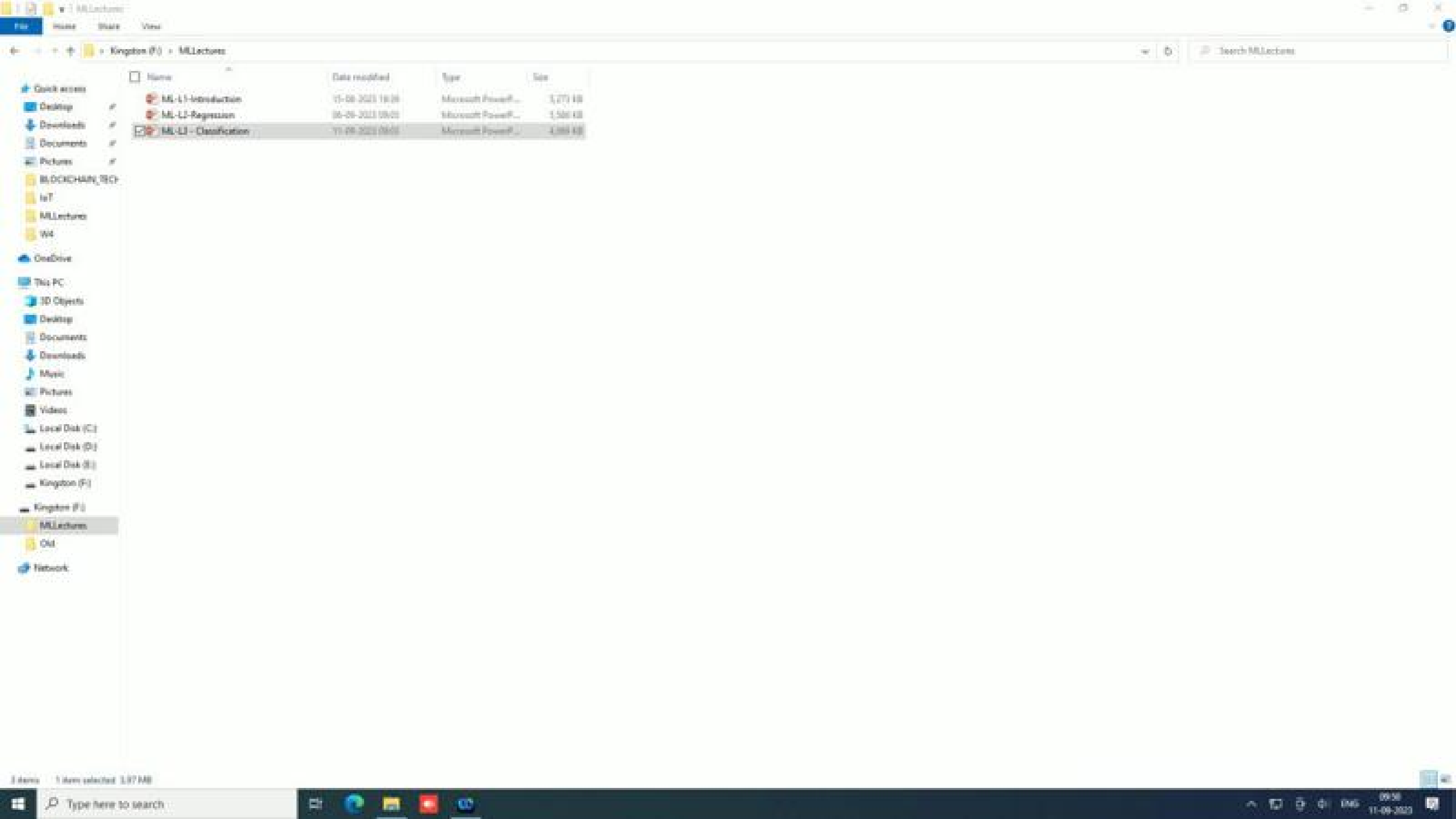
Projection of data from two classes onto various lines

Click to add text



BITS Pilani, Hyderabad Campus

Click to add notes


File
Home
Share
View
Kingston (F:) > MLLectures
Name
Date modified
Type
Size
ML-L1-Introduction
ML-L2-Regression
ML-L3 - Classification
Quick access
Desktop
Downloads
Documents
Pictures
BLOCKCHAIN_TECH
IoT
MLLectures
OneDrive
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Local Disk (D:)
Local Disk (E:)
Kingston (F:)
Kingston (F:)
MLLectures
Old
Network
Type here to search
ENG