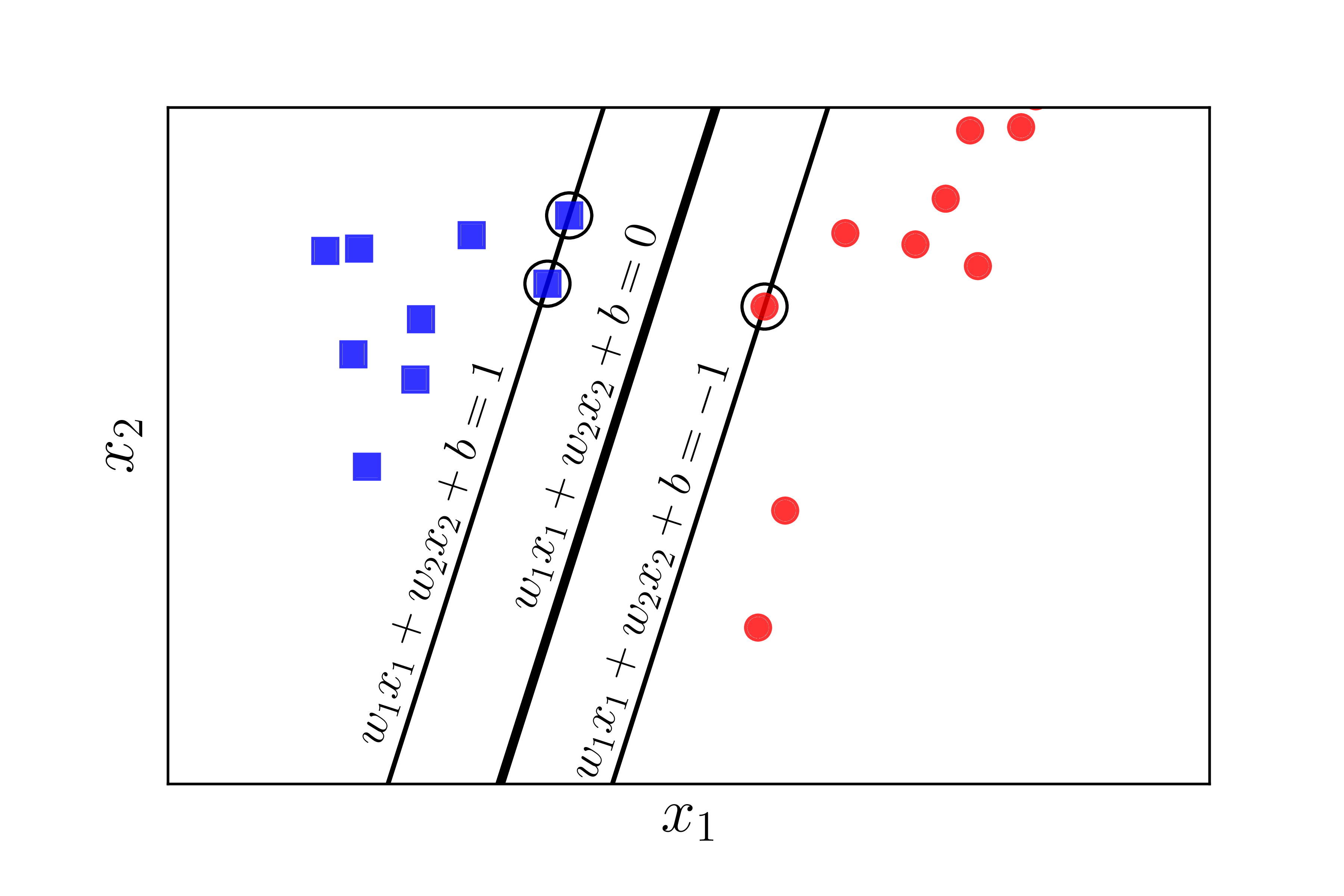
**Report Week 2**

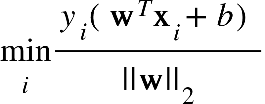
**Exercise 6**

**Introduction to SVM:**

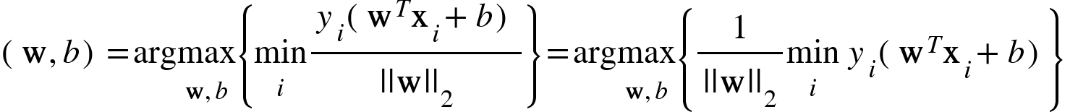
Problem: {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>y</mi><mo>&#x2208;</mo><mfenced open=\"{\" close=\"}\"><mrow><mo>-</mo><mn>1</mn><mo>,</mo><mn>1</mn></mrow></mfenced></mstyle></math>"} ; {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>x</mi><mo>&#x2208;</mo><msup><mi mathvariant=\"normal\">&#x211D;</mi><mi>n</mi></msup></mstyle></math>"}; {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi mathvariant=\"bold\">w</mi><mo mathvariant=\"bold\">,</mo><mi mathvariant=\"bold\">b</mi></mstyle></math>"} : parameters



First, we need find the nearest data point to the hyper plane:



Second, we need to find the best parameter {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>w</mi><mo>,</mo><mi>b</mi><mo>&#xA0;</mo></mstyle></math>"}of hyper-plane to maximize the distance of the nearest point:



**Kernel SVM:**

Kernel SVM used for non-linearly separable data. For example:

A diagram of a line with dots and crosses

Description automatically generatedA green and yellow dotted pattern

Description automatically generated

Linearly Separable Non-linear separable

In this case, the linear plane can’t separate the non-linearly separable data. We use Kernel to transform the initial space of data into new space to easily separate it. For example:

A diagram of a graph

Description automatically generated A diagram of a cone with arrows and points

Description automatically generated

We need use kernal function to transform the data instead of a direct transformation function to reduce the memory and the productivity of computing. For example:

We get a direct transformation here: {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi mathvariant=\"bold\">x</mi><mo>=</mo><mo>[</mo><msub><mi>x</mi><mn>1</mn></msub><mo>,</mo><msub><mi>x</mi><mn>2</mn></msub><msup><mo>]</mo><mi>T</mi></msup><mo>&#xA0;</mo><mo>&#x2192;</mo><mi>&#x3A6;</mi><mo>(</mo><mi mathvariant=\"bold\">x</mi><mo>)</mo><mo>=</mo><mo>[</mo><mn>1</mn><mo>,</mo><msqrt><mn>2</mn></msqrt><msub><mi>x</mi><mn>1</mn></msub><mo>,</mo><msqrt><mn>2</mn></msqrt><msub><mi>x</mi><mn>2</mn></msub><mo>,</mo><msubsup><mi>x</mi><mn>1</mn><mn>2</mn></msubsup><mo>,</mo><msqrt><mn>2</mn></msqrt><msub><mi>x</mi><mn>1</mn></msub><msub><mi>x</mi><mn>2</mn></msub><mo>,</mo><msubsup><mi>x</mi><mn>2</mn><mn>2</mn></msubsup><msup><mo>]</mo><mi>T</mi></msup></mstyle></math>"}

In fact, the data can get higher dimension that the process of computing is too slow.

The Kernel is the better choice for the transformation:

{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>&#x3A6;</mi><mo>(</mo><mi mathvariant=\"bold\">x</mi><msup><mo>)</mo><mi>T</mi></msup><mi>&#x3A6;</mi><mo>(</mo><mi mathvariant=\"bold\">z</mi><mo>)</mo><mo>=</mo><mo>[</mo><mn>1</mn><mo>,</mo><msqrt><mn>2</mn></msqrt><msub><mi>x</mi><mn>1</mn></msub><mo>,</mo><msqrt><mn>2</mn></msqrt><msub><mi>x</mi><mn>2</mn></msub><mo>,</mo><msubsup><mi>x</mi><mn>1</mn><mn>2</mn></msubsup><mo>,</mo><msqrt><mn>2</mn></msqrt><msub><mi>x</mi><mn>1</mn></msub><msub><mi>x</mi><mn>2</mn></msub><mo>,</mo><msubsup><mi>x</mi><mn>2</mn><mn>2</mn></msubsup><mo>]</mo><mo>[</mo><mn>1</mn><mo>,</mo><msqrt><mn>2</mn></msqrt><msub><mi>z</mi><mn>1</mn></msub><mo>,</mo><msqrt><mn>2</mn></msqrt><msub><mi>z</mi><mn>2</mn></msub><mo>,</mo><msubsup><mi>z</mi><mn>1</mn><mn>2</mn></msubsup><mo>,</mo><msqrt><mn>2</mn></msqrt><msub><mi>z</mi><mn>1</mn></msub><msub><mi>z</mi><mn>2</mn></msub><mo>,</mo><msubsup><mi>z</mi><mn>2</mn><mn>2</mn></msubsup><msup><mo>]</mo><mi>T</mi></msup><mspace linebreak=\"newline\"/><mo>=</mo><mn>1</mn><mo>+</mo><mn>2</mn><msub><mi>x</mi><mn>1</mn></msub><msub><mi>z</mi><mn>1</mn></msub><mo>+</mo><mn>2</mn><msub><mi>x</mi><mn>2</mn></msub><msub><mi>z</mi><mn>2</mn></msub><mo>+</mo><msubsup><mi>x</mi><mn>1</mn><mn>2</mn></msubsup><msubsup><mi>x</mi><mn>2</mn><mn>2</mn></msubsup><mo>+</mo><mn>2</mn><msub><mi>x</mi><mn>1</mn></msub><msub><mi>z</mi><mn>1</mn></msub><msub><mi>x</mi><mn>2</mn></msub><msub><mi>z</mi><mn>2</mn></msub><mo>+</mo><msubsup><mi>x</mi><mn>2</mn><mn>2</mn></msubsup><msubsup><mi>z</mi><mn>2</mn><mn>2</mn></msubsup><mo>=</mo><mo>(</mo><mn>1</mn><mo>+</mo><msub><mi>x</mi><mn>1</mn></msub><msub><mi>z</mi><mn>1</mn></msub><mo>+</mo><msub><mi>x</mi><mn>2</mn></msub><msub><mi>z</mi><mn>2</mn></msub><msup><mo>)</mo><mn>2</mn></msup><mo>=</mo><mo>(</mo><mn>1</mn><mo>+</mo><msup><mi mathvariant=\"bold\">x</mi><mi>T</mi></msup><mi mathvariant=\"bold\">z</mi><msup><mo>)</mo><mn>2</mn></msup><mo>=</mo><mi>k</mi><mo>(</mo><mi mathvariant=\"bold\">x</mi><mo>,</mo><mi mathvariant=\"bold\">z</mi><mo>)</mo></mstyle></math>"}

This is polynomial kernel with the degree of 2.

We get the table of the popular kernel:

A white sheet with red text

Description automatically generated

In this exercise, we used the Gaussian Kernel for the transformation of data.

**SVM for Spam Classification:**

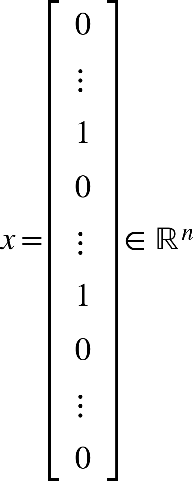
Vectorization an email:

We have a dictionary including 1899 words.

Vectorizing an email to an 1899-dimension vector.

That is, {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi>x</mi><mi>i</mi></msub><mo>=</mo><mn>1</mn></mstyle></math>"} if the i-th word is in the email and {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi>x</mi><mn>0</mn></msub><mo>=</mo><mn>0</mn></mstyle></math>"} if the i-th word is not present in the email.

So that we get a binary n-dimension vectorized email:{"mathml":"<math xmlns=\"http://www.w3.org/1998/Math/MathML\" style=\"font-family:stix;font-size:16px;\"/>"}

{"mathml":"<math xmlns=\"http://www.w3.org/1998/Math/MathML\" style=\"font-family:stix;font-size:16px;\"/>"}

Finally, the spam classification become the problem:

Input:4000 samples vectorized email {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi>X</mi><mi>i</mi></msub><mo>&#x2208;</mo><msup><mi mathvariant=\"normal\">&#x211D;</mi><mn>1899</mn></msup></mstyle></math>"}

Output: 4000 samples {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi>y</mi><mi>i</mi></msub><mo>&#x2208;</mo><mfenced open=\"{\" close=\"}\"><mrow><mn>0</mn><mo>,</mo><mn>1</mn></mrow></mfenced></mstyle></math>"} with 1 means spam and 0 means non-spam.

**Exercise 7**

**Introduction to K-mean Clustering:**

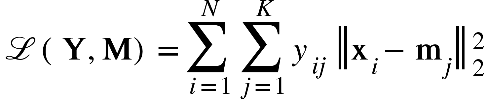
K-means clustering is a popular unsupervised machine learning algorithm used for clustering a given dataset {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>X</mi></mstyle></math>"} into {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>Y</mi></mstyle></math>"} classes.

Input: {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi mathvariant=\"bold\">X</mi><mo>=</mo><mo>[</mo><msub><mi mathvariant=\"bold\">x</mi><mn>1</mn></msub><mo>,</mo><msub><mi mathvariant=\"bold\">x</mi><mn>2</mn></msub><mo>,</mo><mo>&#x2026;</mo><mo>,</mo><msub><mi mathvariant=\"bold\">x</mi><mi>N</mi></msub><mo>]</mo><mo>&#x2208;</mo><msup><mi mathvariant=\"normal\">&#x211D;</mi><mrow><mi>d</mi><mo>&#xD7;</mo><mi>N</mi></mrow></msup></mstyle></math>"} with {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>d</mi></mstyle></math>"} is the dimension of each data point and {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>N</mi></mstyle></math>"} is the number of data points.

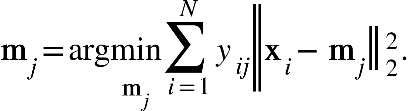
Output: We need find the cluster {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi mathvariant=\"bold\">m</mi><mn>1</mn></msub><mo>,</mo><msub><mi mathvariant=\"bold\">m</mi><mn>2</mn></msub><mo>,</mo><mo>&#x2026;</mo><mo>,</mo><msub><mi mathvariant=\"bold\">m</mi><mi>K</mi></msub><mo>&#x2208;</mo><msup><mi mathvariant=\"normal\">&#x211D;</mi><mrow><mi>d</mi><mo>&#xD7;</mo><mn>1</mn></mrow></msup></mstyle></math>"} and the label {"mathml":"<math xmlns=\"http://www.w3.org/1998/Math/MathML\" style=\"font-family:stix;font-size:16px;\"><mi mathvariant=\"bold\">Y</mi><mo>=</mo><mo>[</mo><msub><mi mathvariant=\"bold\">y</mi><mn>1</mn></msub><mo>;</mo><msub><mi mathvariant=\"bold\">y</mi><mn>2</mn></msub><mo>;</mo><mo>&#x2026;</mo><mo>;</mo><msub><mi mathvariant=\"bold\">y</mi><mi>N</mi></msub><mo>]</mo></math>"}{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"/></math>"}of input

If he label {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi mathvariant=\"bold\">y</mi><mi>i</mi></msub><mo>=</mo><mo>[</mo><msub><mi>y</mi><mrow><mi>i</mi><mn>1</mn></mrow></msub><mo>,</mo><msub><mi>y</mi><mrow><mi>i</mi><mn>2</mn></mrow></msub><mo>,</mo><mo>&#x2026;</mo><mo>,</mo><msub><mi>y</mi><mrow><mi>i</mi><mi>K</mi></mrow></msub><mo>]</mo></mstyle></math>"} is assigned to cluster {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>k</mi></mstyle></math>"} , {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi>y</mi><mrow><mi>i</mi><mi>k</mi></mrow></msub><mo>=</mo><mn>1</mn></mstyle></math>"} and {"mathml":"<math xmlns=\"http://www.w3.org/1998/Math/MathML\" style=\"font-family:stix;font-size:16px;\"><msub><mi>y</mi><mrow><mi>i</mi><mi>j</mi></mrow></msub><mo>=</mo><mn>0</mn><mo>,</mo><mo>&#x2200;</mo><mi>j</mi><mo>&#x2260;</mo><mi>k</mi></math>"}{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"/></math>"}.It’s like one-hot encoding.

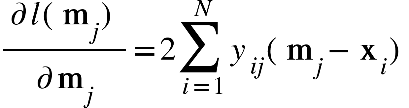
We formally get a Loss function for clustering algorithm:



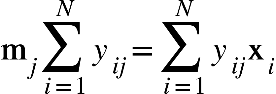
So we need find the best clusters {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>M</mi></mstyle></math>"} to minimize the Loss function {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mo mathvariant=\"script\">L</mo></mstyle></math>"}:

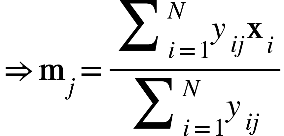


Have {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>l</mi><mo>(</mo><msub><mi mathvariant=\"bold\">m</mi><mi>j</mi></msub><mo>)</mo></mstyle></math>"} is function inside argmin, we derivative {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>l</mi><mo>(</mo><msub><mi mathvariant=\"bold\">m</mi><mi>j</mi></msub><mo>)</mo></mstyle></math>"}to {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi>m</mi><mi>j</mi></msub></mstyle></math>"} :

{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"/></math>"}

With the derivative equal zero, we get the next equation:





The simple indication: {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi mathvariant=\"bold\">m</mi><mi>j</mi></msub></mstyle></math>"} is the average of the data points of cluster {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>j</mi></mstyle></math>"}

**The general algorithm of K-mean Clustering:**

Step 1: Initializing {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>K</mi></mstyle></math>"} clusters

Step 2: Assign each data point into closest center

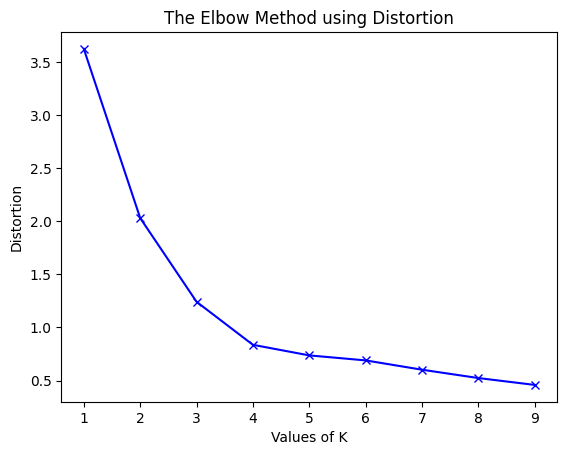
Step 3: If the data points of each class have not changed , The algorithm’s stopped

Step 4: Update the center of each cluster by calculating the average of all data points in step 2

Step 5: Back to Step 2

Elbow curve:

But we don’t know what the best {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>K</mi></mstyle></math>"} value is . Elbow curve is proposed for us to find the best {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>K</mi></mstyle></math>"} value



**Unsupervised segmentation application:**

K-means clustering is used as an unsupervised segmentation

Input: Colour Image

A person with a red hood

Description automatically generated

Output: Segmented Image

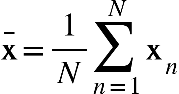
A person's face with red background

Description automatically generated

**Introduction to Principal Component Analysis:**

PCA (Principal Component Analysis) is a statistical and mathematical method used for dimensionality reduction of data. It helps to identify the most important principal components in the data while retaining a significant amount of information.

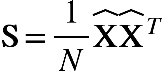
Step 1: Find mean of all data points



Step 2: Subtract each data point to mean

{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mover><mi mathvariant=\"bold\">x</mi><mo>^</mo></mover><mi>n</mi></msub><mo>=</mo><msub><mi mathvariant=\"bold\">x</mi><mi>n</mi></msub><mo>-</mo><mover><mi mathvariant=\"bold\">x</mi><mo>&#xAF;</mo></mover></mstyle></math>"}

Step 3: Compute covariance matrix

{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"/></math>"}

Step 4: Compute eigenvalues and eigenvectors of covariance matrix {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>S</mi></mstyle></math>"}, sorted in decreasing order of eigenvalues

Step 5: Find the {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>K</mi></mstyle></math>"} couple of eigenvalues and eigenvectors to build orthogonal system {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi>U</mi><mi>k</mi></msub></mstyle></math>"}

Step 6: Projection the normalized {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mover><mi mathvariant=\"bold\">X</mi><mo>^</mo></mover></mstyle></math>"} into new subspace

{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi mathvariant=\"bold\">Z</mi><mo>=</mo><msubsup><mi mathvariant=\"bold\">U</mi><mi>K</mi><mi>T</mi></msubsup><mover><mi mathvariant=\"bold\">X</mi><mo>^</mo></mover></mstyle></math>"}

A diagram of a computer procedure

Description automatically generated

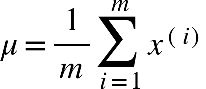
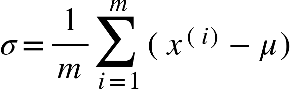
**Exercise 8**

**Introduction to Anomaly Detection:**

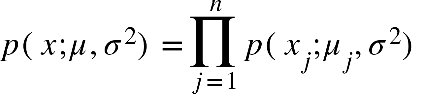
{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>x</mi><mo>&#x2208;</mo><mo mathvariant=\"double-struck\">(</mo><mi>R</mi><mo>)</mo></mstyle></math>","origin":"MathType for Microsoft Add-in"} is a distributed Gaussian with mean 𝜇 and variance 𝜎 :

{"mathml":"<math xmlns=\"http://www.w3.org/1998/Math/MathML\" style=\"font-family:stix;font-size:16px;\"><mi>x</mi><mo>~</mo><mi mathvariant=\"script\">N</mi><mo>(</mo><mi>&#x3BC;</mi><mo>,</mo><msup><mi>&#x3C3;</mi><mn>2</mn></msup><mo>)</mo></math>","origin":"MathType for Microsoft Add-in"}

We fit the parameters:

;

We get the PDF ( Probability Density Function):



With {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>n</mi></mstyle></math>","origin":"MathType for Microsoft Add-in"} is the number of features and {"mathml":"<math xmlns=\"http://www.w3.org/1998/Math/MathML\" style=\"font-family:stix;font-size:16px;\"><mi>m</mi></math>","origin":"MathType for Microsoft Add-in"} is the number of samples

A data point is anomaly if {"mathml":"<math xmlns=\"http://www.w3.org/1998/Math/MathML\" style=\"font-family:stix;font-size:16px;\"><mi>p</mi><mo>(</mo><mi>x</mi><mo>)</mo><mo>&lt;</mo><mi>&#x3F5;</mi></math>","origin":"MathType for Microsoft Add-in"}

The example of probability distribution of each features :

A diagram of a function

Description automatically generated

For example, the data points’s out of the yellow border coutour is anomaly:

A computer screen shot of a diagram

Description automatically generated

The yellow border coutour is respresented as {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>&#x3F5;</mi></mstyle></math>","origin":"MathType for Microsoft Add-in"}

**F1 Score:**

Precision and Recall:

We suppose that there are 2 categories , Positive and Negative

TP is True Positive

FP is Fail Positive

TN is True Negative

FN is Fail Negative

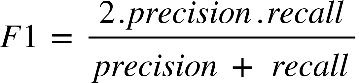


The data points inside circle border is predicted as positive

Precision is defined as the proportion between True positive (TP) to clasified positive (TP + FP)

Recall is defined as the proportion between True positive (TP) to real positive (TP + FN)

So , F1 score is formulated as:



**Recommender System:**

A recommender system is a type of information filtering system or algorithm that predicts and suggests items or content to users based on their preferences, historical behavior, or characteristics.

We formulate the recommender system based on user’s previous rating on movies :

-{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>r</mi><mfenced><mrow><mi>i</mi><mo>,</mo><mi>j</mi></mrow></mfenced><mo>=</mo><mn>1</mn></mstyle></math>","origin":"MathType for Microsoft Add-in"} if user {"mathml":"<math xmlns=\"http://www.w3.org/1998/Math/MathML\" style=\"font-family:stix;font-size:16px;\"><mi>j</mi></math>","origin":"MathType for Microsoft Add-in"} has rated movie {"mathml":"<math xmlns=\"http://www.w3.org/1998/Math/MathML\" style=\"font-family:stix;font-size:16px;\"><mi>i</mi></math>","origin":"MathType for Microsoft Add-in"}

-{"mathml":"<math xmlns=\"http://www.w3.org/1998/Math/MathML\" style=\"font-family:stix;font-size:16px;\"><msup><mi>y</mi><mfenced><mrow><mi>i</mi><mo>,</mo><mi>j</mi></mrow></mfenced></msup></math>","origin":"MathType for Microsoft Add-in"} : rating by user {"mathml":"<math xmlns=\"http://www.w3.org/1998/Math/MathML\" style=\"font-family:stix;font-size:16px;\"><mi>j</mi></math>","origin":"MathType for Microsoft Add-in"} on movie {"mathml":"<math xmlns=\"http://www.w3.org/1998/Math/MathML\" style=\"font-family:stix;font-size:16px;\"><mi>i</mi></math>","origin":"MathType for Microsoft Add-in"}

-{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msup><mi>&#x3B8;</mi><mfenced><mi>j</mi></mfenced></msup></mstyle></math>","origin":"MathType for Microsoft Add-in"} : parameter vector for user {"mathml":"<math xmlns=\"http://www.w3.org/1998/Math/MathML\" style=\"font-family:stix;font-size:16px;\"><mi>j</mi></math>","origin":"MathType for Microsoft Add-in"}

-{"mathml":"<math xmlns=\"http://www.w3.org/1998/Math/MathML\" style=\"font-family:stix;font-size:16px;\"><msup><mi>x</mi><mfenced><mi>i</mi></mfenced></msup></math>","origin":"MathType for Microsoft Add-in"}: feature vector for movie {"mathml":"<math xmlns=\"http://www.w3.org/1998/Math/MathML\" style=\"font-family:stix;font-size:16px;\"><mi>i</mi></math>","origin":"MathType for Microsoft Add-in"}

-For user {"mathml":"<math xmlns=\"http://www.w3.org/1998/Math/MathML\" style=\"font-family:stix;font-size:16px;\"><mi>j</mi></math>","origin":"MathType for Microsoft Add-in"} , movie {"mathml":"<math xmlns=\"http://www.w3.org/1998/Math/MathML\" style=\"font-family:stix;font-size:16px;\"><mi>i</mi></math>","origin":"MathType for Microsoft Add-in"} , predicting rating {"mathml":"<math xmlns=\"http://www.w3.org/1998/Math/MathML\" style=\"font-family:stix;font-size:16px;\"><mo>(</mo><msup><mi>&#x3B8;</mi><mrow><mo>(</mo><mi>j</mi><mo>)</mo></mrow></msup><msup><mo>)</mo><mi>T</mi></msup><mo>(</mo><msup><mi>x</mi><mrow><mo>(</mo><mi>i</mi><mo>)</mo></mrow></msup><mo>)</mo></math>","origin":"MathType for Microsoft Add-in"}

Our Target is that optimizing this function:

