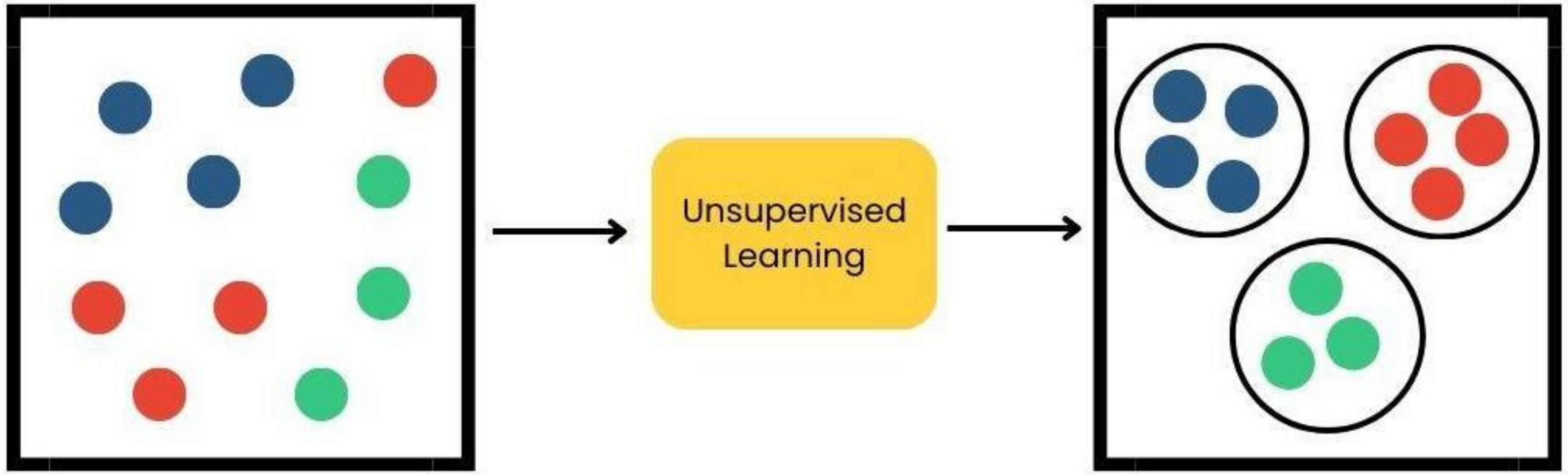
COMP30027 MACHINE LEARNING TUTORIAL

Workshop - 11

Unsupervised and Semi-Supervised Learning

What is Unsupervised Learning?



No Ground Truth

Unsupervised learning algorithms learn from data that has no labels.

Pattern Discovery

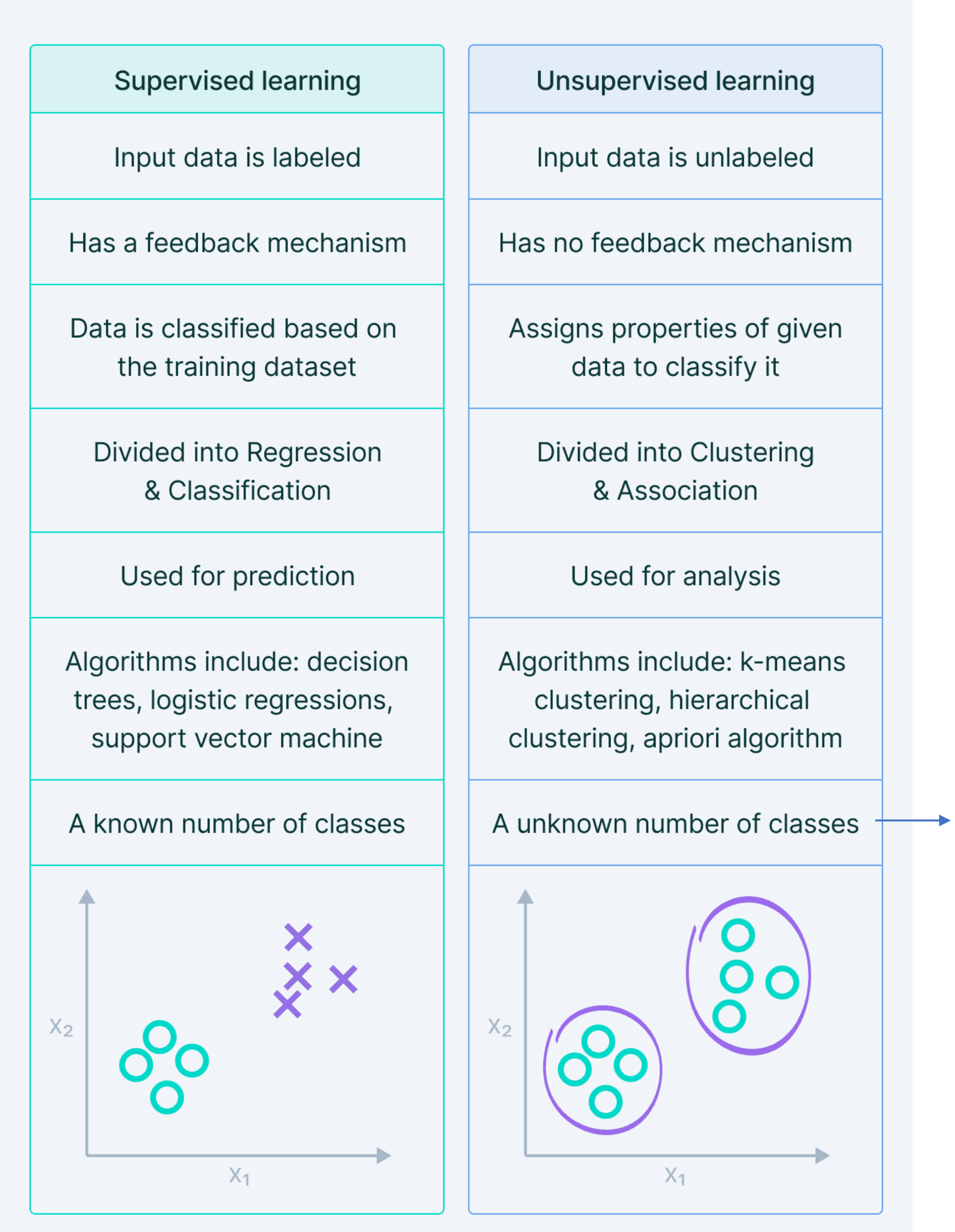
Identifies structures and relationships autonomously

Hidden Insights

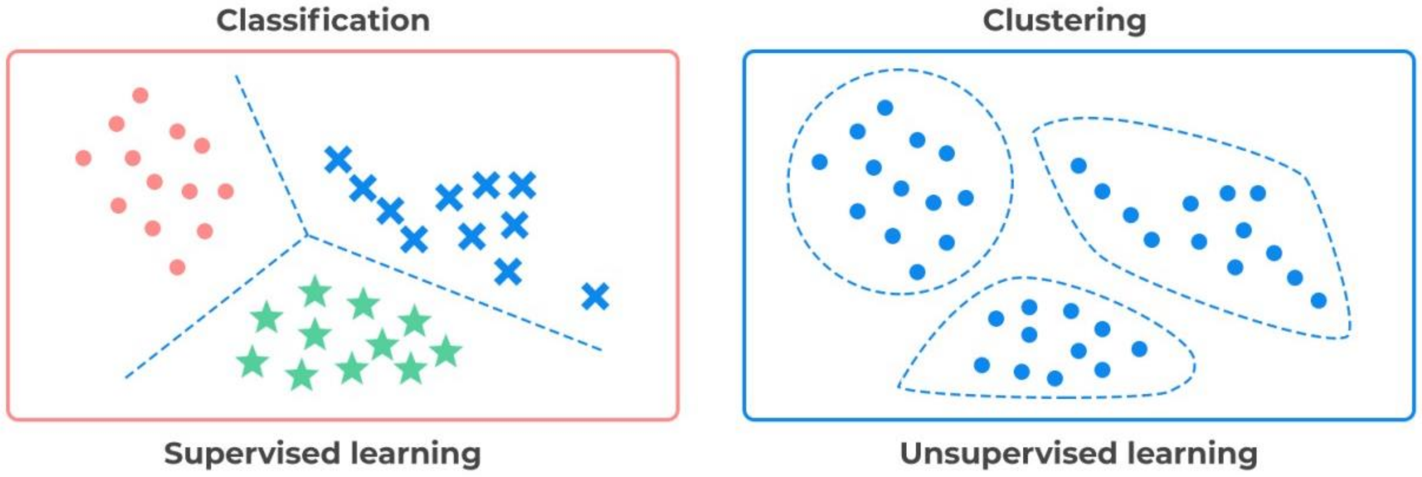
Processes raw data into meaningful representations

https://www.bombaysoftwares.com/blog/introduction-to-unsupervised-learning

Supervised vs Unsupervised Learning



Supervised learning	Unsupervised learning
Input data is labeled	Input data is unlabeled
Has a feedback mechanism	Has no feedback mechanism
Data is classified based on the training dataset	Assigns properties of given data to classify it
Divided into Regression & Classification	Divided into Clustering & Association
Used for prediction	Used for analysis
Algorithms include: decision trees, logistic regressions, support vector machine	Algorithms include: k-means clustering, hierarchical clustering, apriori algorithm
A known number of classes	A unknown number of classes



But in practice, **we often choose a value for K** (number of clusters)

https://www.linkedin.com/pulse/supervised-vs-unsupervised-learning-whats-difference-smriti-saini/

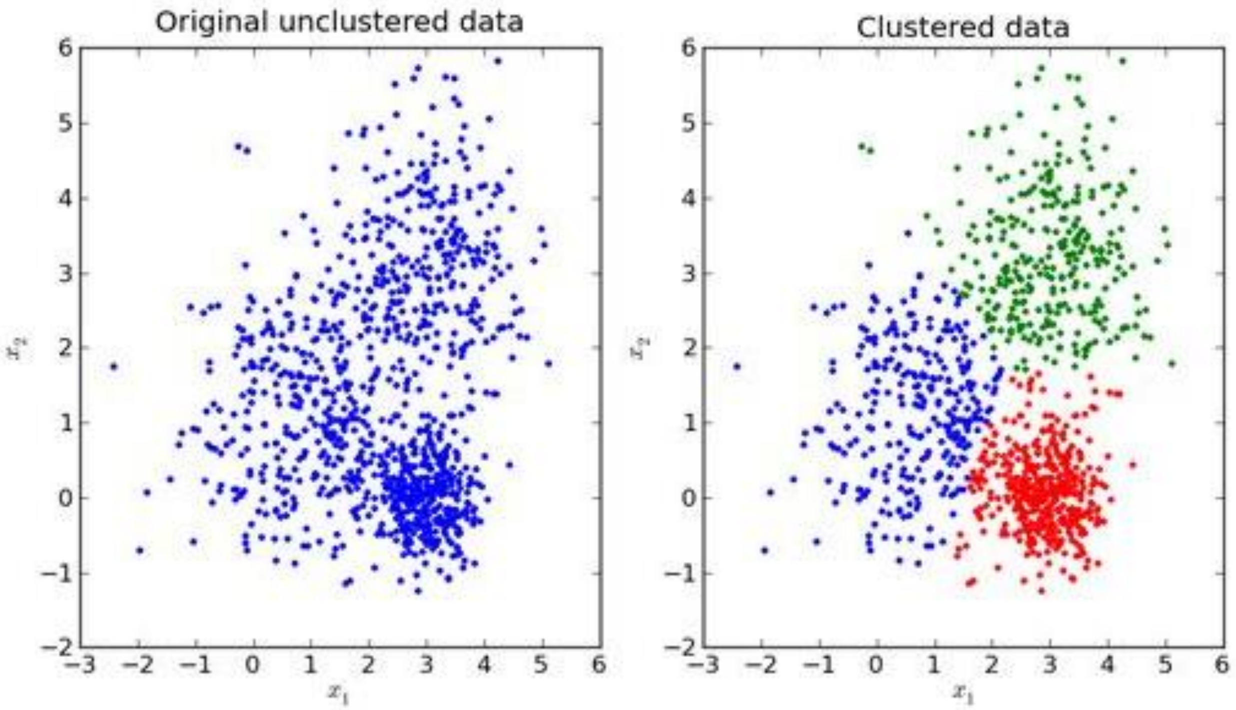
https://www.v7labs.com/blog/supervised-vs-unsupervised-learning

Clustering

What is Clustering?

Clustering is a **type of unsupervised learning** where the algorithm tries to **group data points** such that:

- Points in the **same group (cluster)** are more similar to each other.
- Points in **different clusters** are dissimilar.



K-means

How K-means Works:

1.Choose **K** clusters (e.g., K = 3).

2.Randomly select **K points** as the **initial centroids** (cluster centers).

3.For each instance, compute the distance to each centroid:

   1. Assign each point to the cluster with **nearest centroid**.
   2. Recalculate a new centroid for each cluster (centroid =mean of all instances in the cluster).

4. Go to step 3, repeat until no instances are reassigned.

https://roboticsknowledgebase.com/wiki/math/gaussian-process-gaussian-mixture-model/

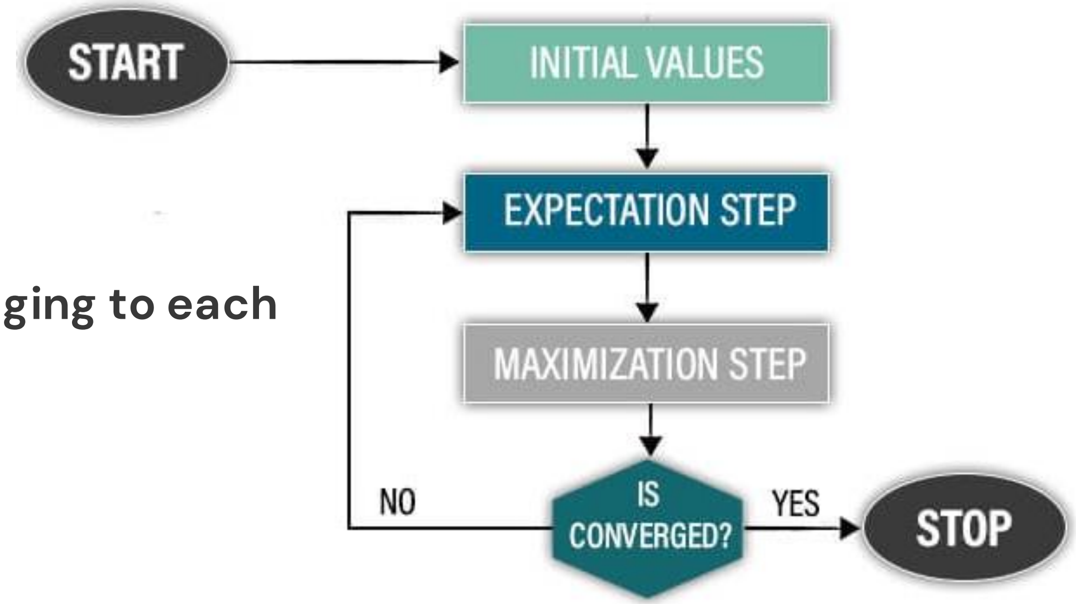
Expectation-Maximisation (EM) Algorithm

EM is an iterative algorithm used in unsupervised machine learning tasks such as clustering. It **is a parameter estimation method.** In other words, the basic idea behind the EM algorithm is to estimate the parameters of a statistical model when some of the data is missing or unobserved. In the case of clustering, the missing data refers to the assignment of data points to clusters.

Initialization

Initialise the algorithm by providing random or uniform initial values for model parameters.

Flowchart of EM Algorithm



Expectation Step

The expectation stage **computes the probabilities of each data point belonging to each cluster** based on the current estimate of the cluster parameters.

Maximization Step

The maximisation step uses the expected value (posterior probability) generated from the expectation step and **updates the model parameter to maximise the likelihood** of the given data

Iteration

The converged step checks if the change in the parameter is below the predetermined threshold or if the maximum number of iterations has reached

https://www.educba.com/em-algorithm-in-machine-learning/

Gaussian Mixture Models (GMM)

The GMM (Gaussian Mixture Model) method is a popular application of the EM algorithm for clustering.

GMM is a probabilistic model that represents data as a mixture of multiple Gaussian distributions, i.e., in the GMM method, each cluster is modelled as a Gaussian distribution with a mean and a covariance matrix. The probability of a data point belonging to a cluster is computed as the probability density of the point under the corresponding Gaussian distribution.

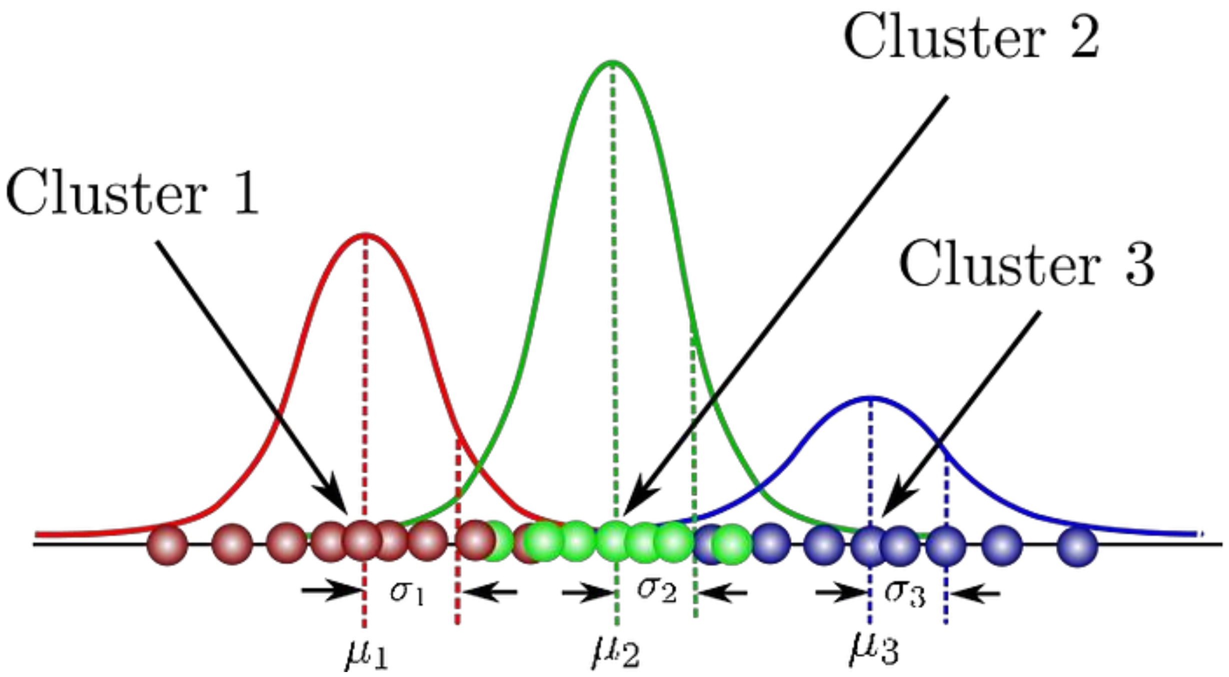
Unlike traditional clustering methods, GMM provides soft assignments, meaning each data point belongs to multiple clusters with varying probabilities.

Since GMM is an application of the EM algorithm, it also has 2 major steps:

In the **Expectation step**, it calculates the **probability of each data point belonging to each Gaussian component or cluster**.

In the **Maximization step**, it updates the **mean and standard deviation to maximise the likelihood of the given data**.

This process repeats until the model converges to a stable solution.



Each cluster is a **Gaussian** (bell curve) with its own mean and variance.

https://medium.com/data-science/gaussian-mixture-models-explained-6986aaf5a95

Q1

What is the logic behind the EM algorithm, when used for clustering?

1. Explain the significance of the "E" step, and the "M" step.
2. What happens in the "E" and "M" steps of the GMM method?

1. EM Algorithm Steps

1. **Initialization**:

   Start with **initial guesses** (random or uniform) for model parameters (e.g., cluster means, variances, priors).

2. **E-step (Expectation)**:

   Estimate the **missing data** (e.g., cluster posterior probabilities), i.e., in the case of clustering, compute the **probability** that each data point belongs to each cluster.

3. **M-step (Maximization)**:

   Update the model parameters based on the expected value of missing data from the E-step, i.e., in the case of clustering, recompute cluster parameters based on the posterior probability.

4. **Repeat**:

   Alternate between E and M steps until the model **converges**.

2. Gaussian Mixture Model (GMM) with EM

- GMM models the data as a **mixture of multiple Gaussian distributions** (clusters).
- Each cluster has its own:
  - **Mean** (µ)
  - **standard deviation** (σ)

E-step in GMM

- For each data point:
  - Calculate the responsibility (γ) or posterior probability of each Gaussian cluster based on the current µ and σ values using **Bayes' rule**.
  - This measures *the expected number of data points assigned to each cluster, i.e.,* how likely the data point came from each Gaussian or cluster.

M-step in GMM

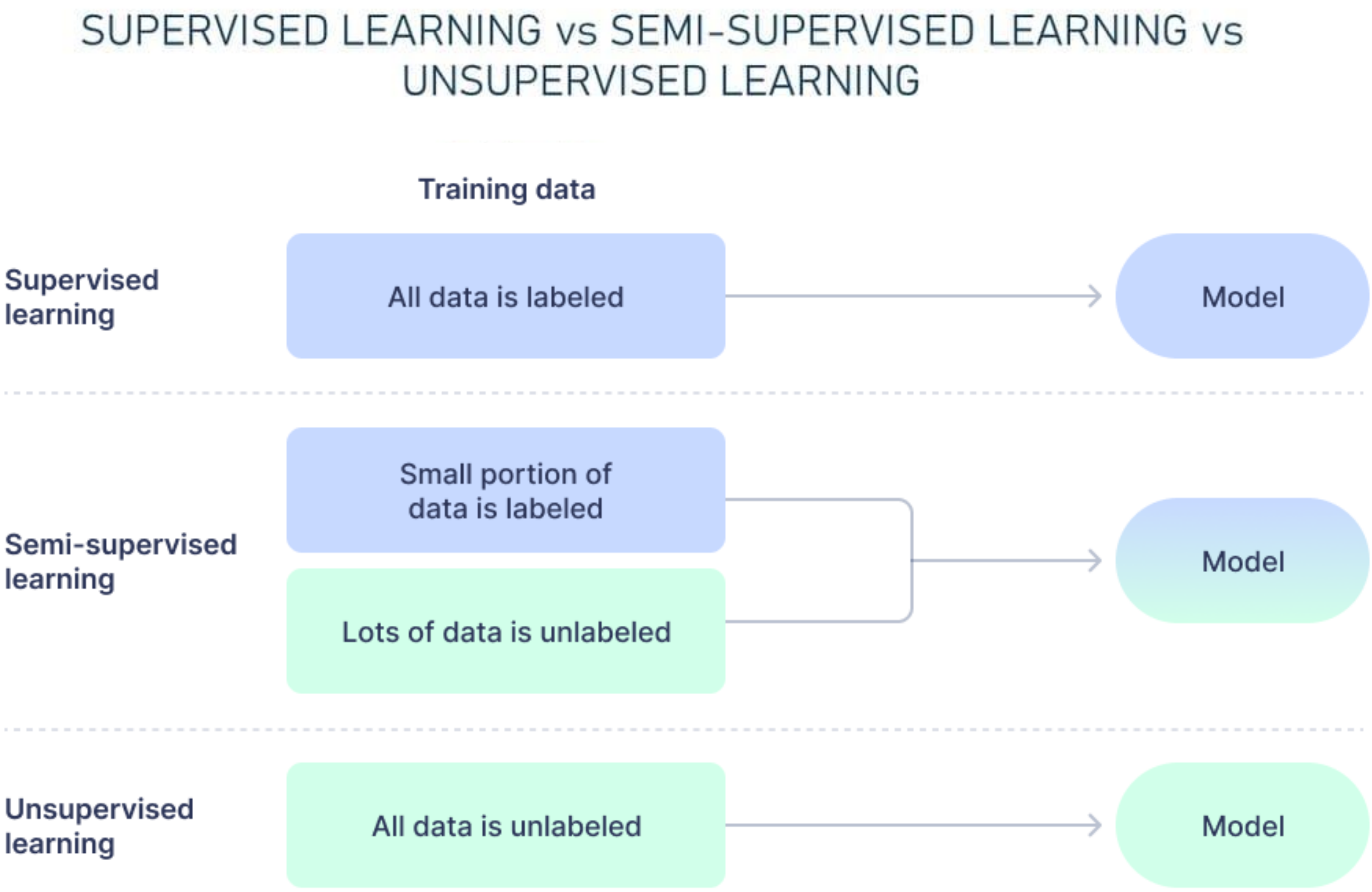
- Update the cluster parameters for each Gaussian cluster based on the corresponding responsibility (γ) value computed in the E-step:
  - **New mean (µ)**: Weighted average of points, using responsibilities.
  - **New standard deviation (σ)**: Spread of points around the mean, weighted by responsibilities.
- The M-step maximizes the likelihood of the data given the expected number of data points assigned to each cluster.

We keep iterating between these two steps until the algorithm converges (e.g., the log likelihood cannot be further improved). The final estimates of the cluster parameters are used to assign each data point to a cluster.

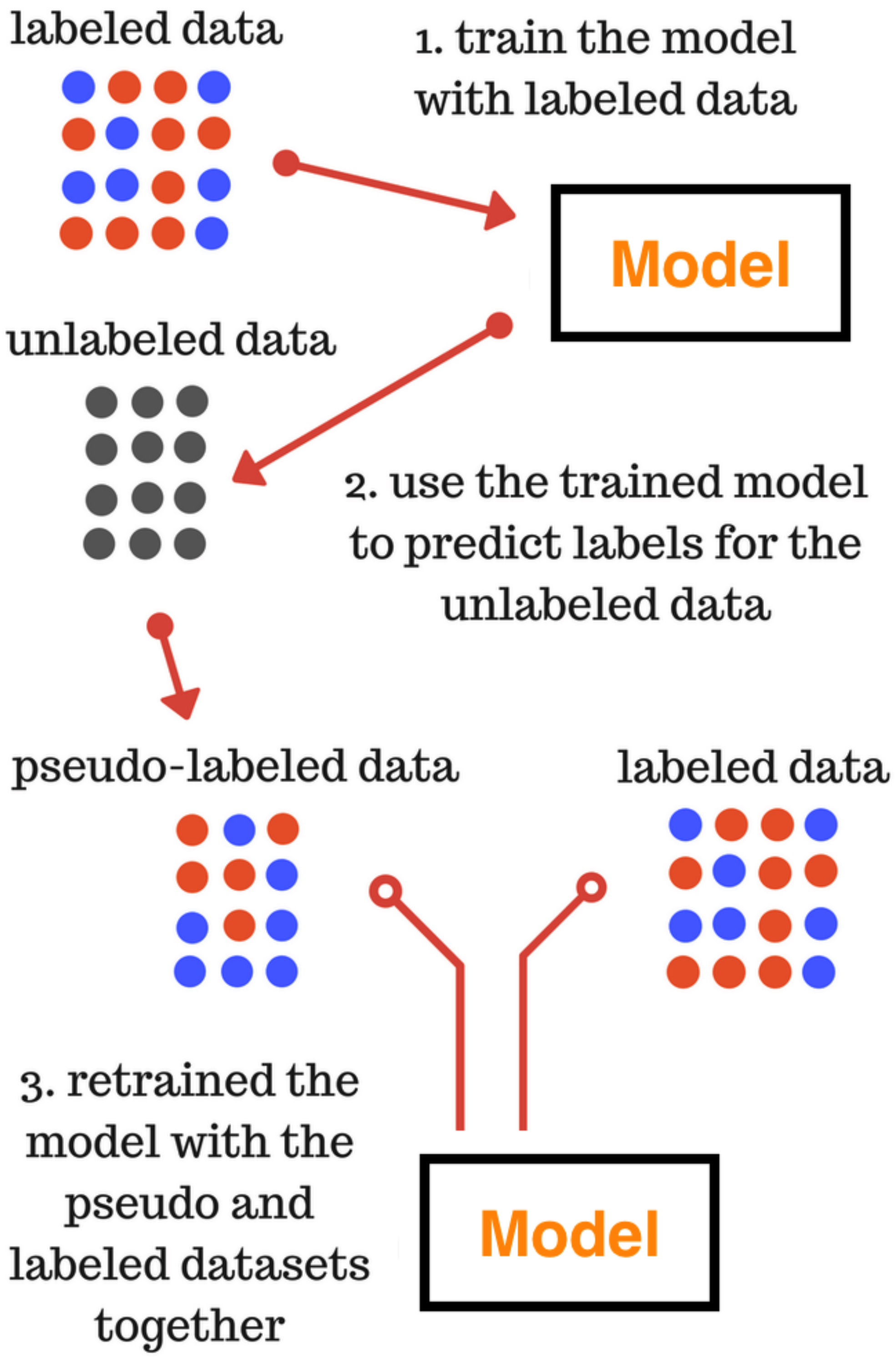
What is Semi-Supervised Learning?

A **semi-supervised algorithm** is a learning method that uses both:

- **Labelled data** (which is limited and costly to get)
- **Unlabelled data** (which is abundant and cheap)



https://quiq.com/blog/semi-supervised-learning-explained-with-examples/


labeled data
1. train the model with labeled data
Model
unlabeled data
2. use the trained model to predict labels for the unlabeled data
pseudo-labeled data
labeled data
3. retrained the model with the pseudo and labeled datasets together
Model

Self-Training: Definition and Steps

Self-training is a wrapper method where a supervised model is trained on a small labelled dataset and then used to assign pseudo-labels to unlabelled data. The most confident predictions are added to the labelled set in an iterative process.

Process:

1.Train a classifier on the small labelled dataset.

2.Use it to predict labels on the unlabelled dataset.

3.Select high-confidence predictions and treat them as true labels (pseudo-labels).

4.Add them to the labelled dataset and retrain the model.

5.Repeat until the stopping criterion is met.

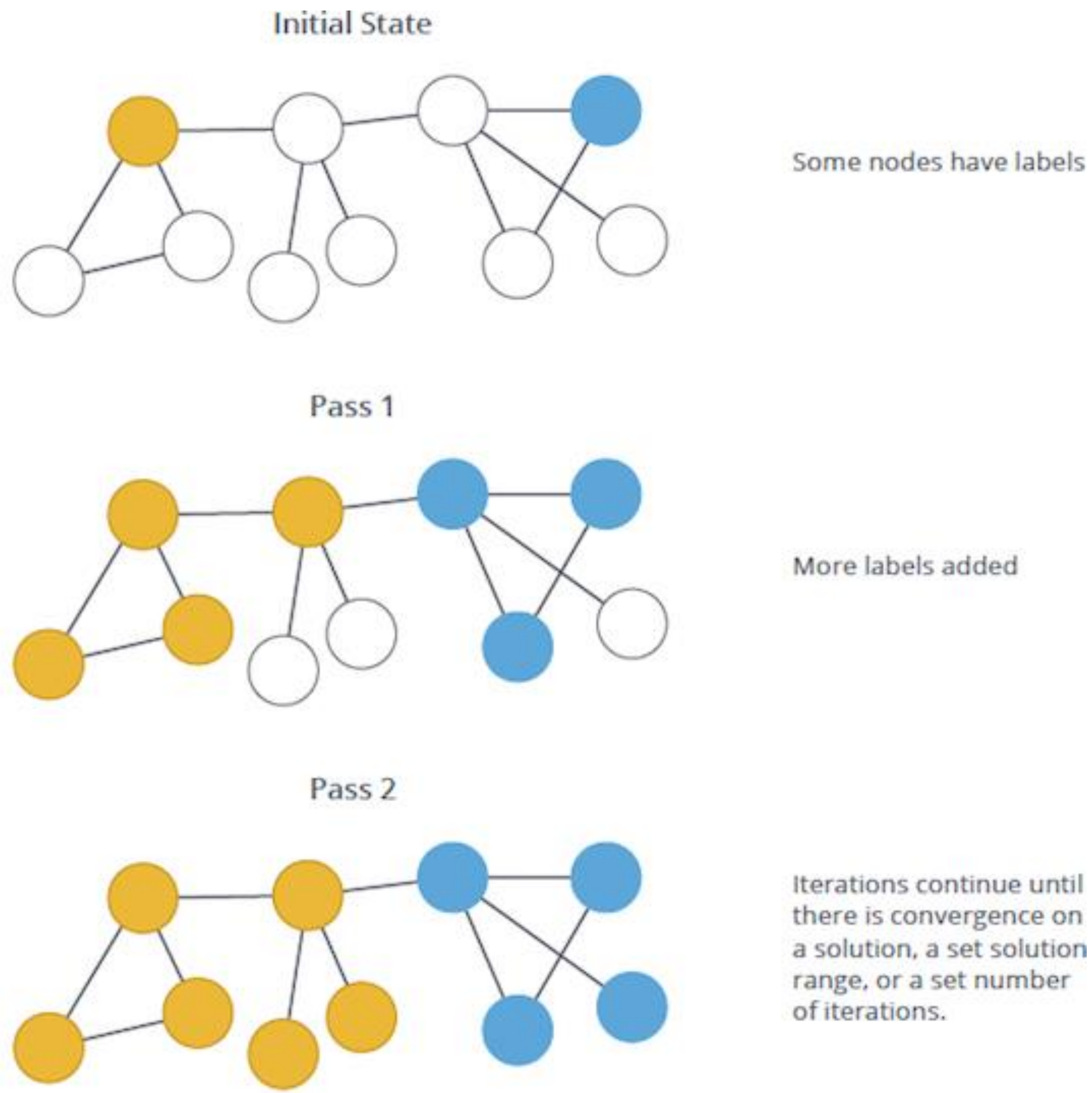
https://jayhey.github.io/semi-supervised%20learning/2017/12/07/semisupervised_self_training/

Self-Training – Case Scenario

Scenario: Email Spam Detection

- You're building a spam classifier.
- You have **500 labelled emails** (spam/not spam) and **50,000 unlabelled emails**.
- You train a logistic regression classifier on the labelled data.
- The model predicts labels for the unlabeled emails.
- You **select only those with >95% confidence** and add them to your labelled set as pseudo-labels.
- Retrain and repeat.

Label Propagation: Definition and Steps



Initial State

Some nodes have labels

Pass 1

More labels added

Pass 2

Iterations continue until there is convergence on a solution, a set solution range, or a set number of iterations.

Label Propagation Algorithm

Label propagation is a graph-based semi-supervised learning method. It treats all samples (labelled + unlabelled) as nodes in a graph and spreads labels from labelled nodes to nearby unlabelled nodes using similarity.

Build Graph

Create similarity connections between data points.

(nodes = samples, edges = similarity)

Assign Labels

Set known labels to labeled nodes

Propagate

Iteratively propagate labels to unlabeled nodes based on similarity and neighbors' labels

Converge

Repeat until stable solution reached

https://neo4j.com/blog/graph-data-science/graph-algorithms-neo4j-label-propagation/

Label Propagation – Case Scenario

- **Scenario: Product Categorisation in E-commerce**

- You have 1,000 products, but only **50 are labelled** with categories like "Electronics", "Clothing", "Toys".

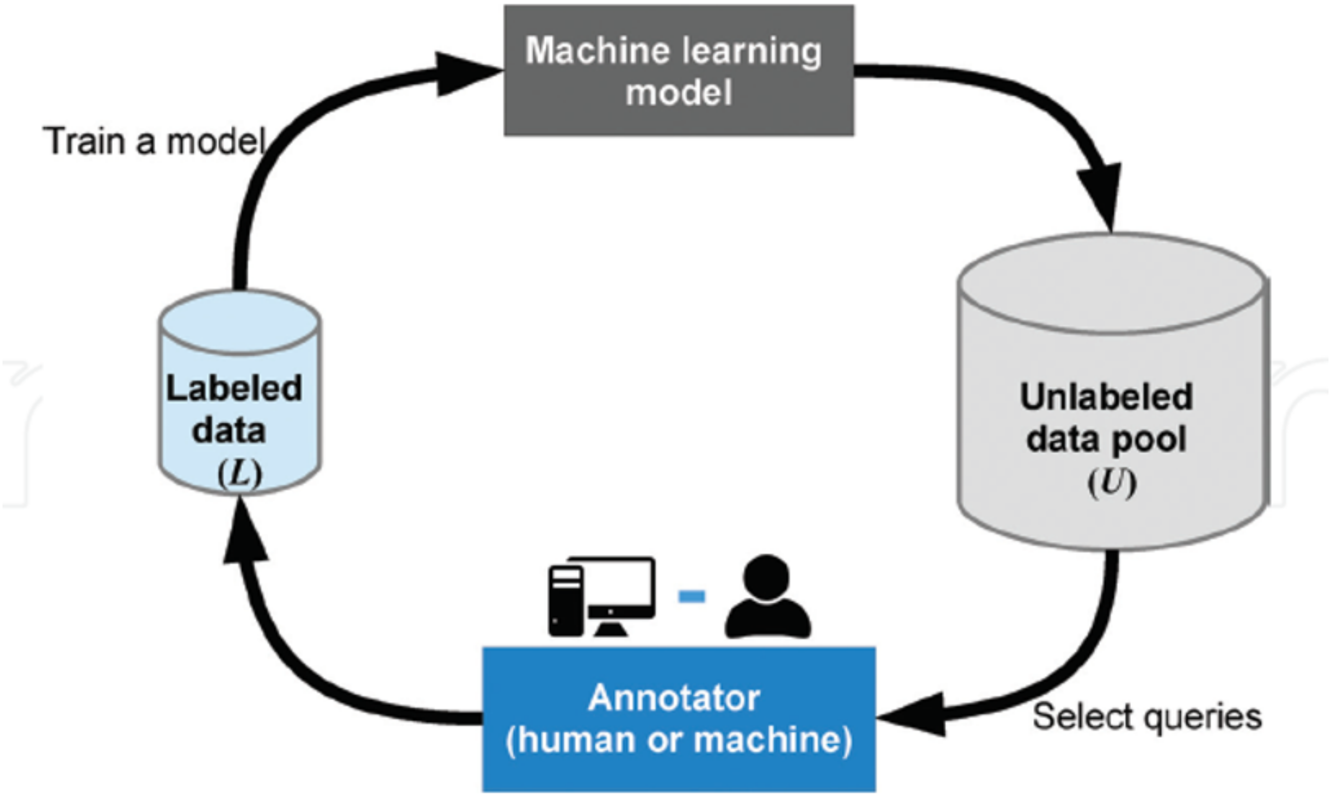
- Each product has a **text description and image**, and you can compute **feature vectors**.

- You build a **similarity graph** (e.g., cosine similarity of text/image features).

- Labels from the known 50 products are **propagated** to the rest of the 950 using label propagation

Active Learning

Active learning is a **human-in-the-loop** approach where the model **actively selects the most informative unlabelled samples** and queries an oracle (usually a human) to label them.



Steps:

1. Train a model on the **initial labelled dataset**.
2. Use the model to identify **high-uncertainty instances** (e.g., low confidence, high entropy).
3. Query an **oracle(** e.g., **human annotator)** to label these specific instances.
4. Add the newly labelled instances to the training set.
5. Retrain the model.
6. Repeat until the convergence condition.

Uncertainty Selection Methods:

- **Uncertainty Sampling**: Select points where the model is least confident.
- **QBC (Query by Committee)**: Use multiple models and pick instances with the most disagreement.

The main assumption of active learning is that instances which are difficult for a model to classify are the most informative for learning

https://doi.org/10.18608/jla.2024.8007

Active Learning – Case Scenario

Scenario: Medical Imaging – Tumour Classification

- You're training a model to detect cancer from MRI scans.
- **Labelling requires a radiologist**, which is **time-consuming and costly**.
- You start with 100 labelled images.
- The model is trained and identifies **unlabelled scans with the highest uncertainty** (e.g., prediction probability near 50%).
- You **ask a radiologist** to label only those scans.
- Add them to your training set and retrain.

Q2

What is the main assumption of self-training? What is the main assumption of active learning?

- *The main assumption of **self-training** is that **similar instances are likely to have the same label**.*
- *A common approach is to find the most similar unlabelled instances to our labelled data and, if the similarity is high enough (better than our defined threshold), give them the same label and add them to the "labelled" training dataset.*

- *The main assumption of **active learning** is that **instances that are difficult for a model to classify are the most informative for learning.***
- *That's why we find the instances that we are most uncertain about (using different methods such as QBC or Uncertainty Sampling) and send them to the human annotator (Oracle).*
- *The assumption here is that having correct labels for these instances will be most helpful for learning the correct class boundaries.*

Query-By-Committee

- Query-By-Committee (QBC) is another popular strategy in active learning in machine learning.
- Train multiple classifiers on a labelled dataset, use each to predict on unlabelled data, and select instances with the highest disagreement between classifiers.
- Disagreement can be measured by entropy.

QBC uses the equation below, which captures vote entropy, to determine the instance that our active learner would select first.

$$x_{VE} = \underset{x}{\operatorname{argmax}}\left(-\sum_{y_i} \frac{V(y_i)}{C} log_2 \frac{V(y_i)}{C}\right)$$

x → *instance,*
y_{i} → *set of possible class labels*
$V(y_i)$ → *number of votes a given label receives from the classifiers*
C → *number of classifiers.*

Q3

1. Describe the rationale and key principles behind the query-by-committee (QBC) algorithm.
2. The table below shows the predicted class labels (A, B, or C) from four classifiers (C1 - C4) for three instances (1-3). Use QBC to determine the instance that an active learner would select first in this scenario.

Instance	C1 pred	C2 pred	C3 pred	C4 pred
1	B	C	A	B
2	B	B	B	B
3	A	C	A	C

1. The goal of active learning is to achieve **high accuracy with as few queries from the oracle as possible**, by selecting the most informative or uncertain examples to query.

   One of the strategies for query sampling is query-by-committee (QBC), where a set of classifiers is trained over a fixed training set, and the **instance that results in the highest disagreement amongst the classifiers is selected for querying.**

   The idea is that the **models will have different strengths and weaknesses,** and by combining their predictions, the **algorithm can leverage their collective intelligence.**

   The rationale behind QBC is that **diversity in the committee is crucial** for achieving better accuracy.

Instance	C1 pred	C2 pred	C3 pred	C4 pred
1	B	C	A	B
2	B	B	B	B
3	A	C	A	C

2. For each instance, we then calculate the total number of votes received by each label class:

Instance	A	B	C
1	1	2	1
2	0	4	0
3	2	0	2

$$x_{VE} = \underset{x}{\operatorname{argmax}}(-\sum_{y_i} \frac{V(y_i)}{C} log_2 \frac{V(y_i)}{C})$$

x → *instance,*

y_i → *set of possible class labels*

$V(y_i)$ → *number of votes a given label receives from the classifiers*

C → *number of classifiers.*

Calculating the vote entropy for each instance yields:

$$instance1 : -(\frac{1}{4}log_2\frac{1}{4} + \frac{2}{4}log_2\frac{2}{4} + \frac{1}{4}log_2\frac{1}{4}) = 1.5$$

$$instance2 : -(\frac{4}{4}log_2\frac{4}{4}) = 0$$

$$instance3 : -(\frac{2}{4}log_2\frac{2}{4} + \frac{2}{4}log_2\frac{2}{4}) = 1$$

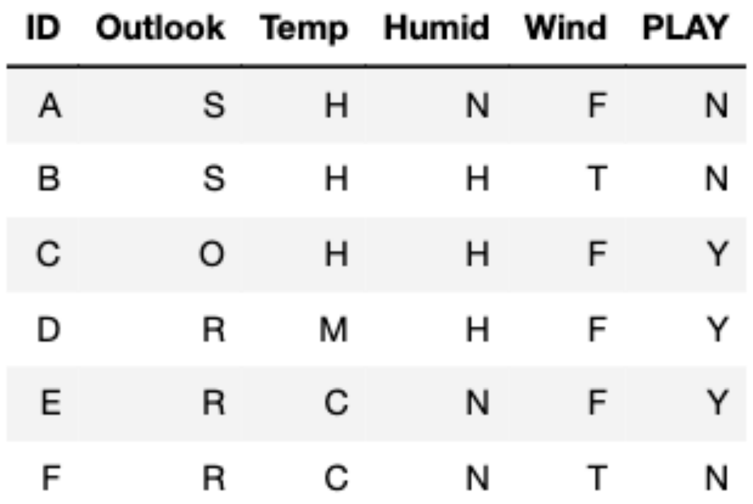
Instance 1 has the highest vote entropy, indicating the greatest disagreement among the four classifiers.

According to QBC, it should be selected for labelling.

Instance 1 may have higher entropy because it possibly lies on the boundary between the three classes; therefore, by querying this instance, we might learn more about the data space.

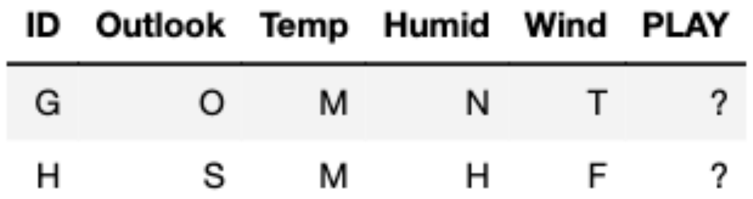
Q4

Consider a naive Bayes model trained using the following familiar weather dataset:



ID	Outlook	Temp	Humid	Wind	PLAY
A	S	H	N	F	N
B	S	H	H	T	N
C	O	H	H	F	Y
D	R	M	H	F	Y
E	R	C	N	F	Y
F	R	C	N	T	N

Suppose that you made additional observations of days and their features. But you don't have the label for the PLAY in these days:



ID	Outlook	Temp	Humid	Wind	PLAY
G	O	M	N	T	?
H	S	M	H	F	?

How could you incorporate this information into your naive Bayes model without manually annotating the labels? If necessary, recompute your model parameters.

The unlabelled instances can be incorporated into the model using self-training as follows:

1. *Train the learner on the currently labelled instances.*
2. *Use the learner to predict the labels of the unlabeled instances.*
3. *Where the learner is very confident, add newly labelled instances to the training set.*
4. *Repeat until all instances are labelled, or no new instances can be labelled confidently.*

For step 1, let's assume we have trained a naive Bayes classifier using Laplace smoothing (alpha = 1). Given this model, the two unlabelled instances will be classified as follows:

How many times value v appears for class c

(α=1)

$$P_i = \frac{x_i + \alpha}{N + \alpha d}$$

How many total instances have class c

No. of levels or distinct values in that Feature

Instance G

$$N{:}P(N)\times P(Outlook{=}O\,|\,N)\times P(Temp{=}M\,|\,N)\times P(Humid{=}N\,|\,N)\times P(Wind{=}T\,|\,N)$$

$$P(N) = \tfrac{3}{6} = \tfrac{1}{2}$$

$$P(O \mid N) = \frac{0+1}{3+3} = \frac{1}{6}$$

$$P(M \mid N) = \frac{0+1}{3+3} = \frac{1}{6}$$

$$P(N \mid N) = \frac{2+1}{3+2} = \frac{3}{5}$$

$$P(T \mid N) = \frac{2+1}{3+2} = \frac{3}{5}$$

$$= \frac{1}{2}(\frac{1}{6})(\frac{1}{6})(\frac{3}{5})(\frac{3}{5}) = 0.005$$

ID	Outlook	Temp	Humid	Wind	PLAY
A	S	H	N	F	N
B	S	H	H	T	N
C	O	H	H	F	Y
D	R	M	H	F	Y
E	R	C	N	F	Y
F	R	C	N	T	N

ID	Outlook	Temp	Humid	Wind	PLAY
G	O	M	N	T	?
H	S	M	H	F	?

Instance G

$$Y: P(Y) \times P(Outlook{=}O \mid Y) \times P(Temp{=}M \mid Y) \times P(Humid{=}N \mid Y) \times P(Wind{=}T \mid Y)$$

$$P(Y) = \tfrac{3}{6} = \tfrac{1}{2}$$

$$P(O \mid Y) = \frac{1+1}{3+3} = \frac{2}{6}$$

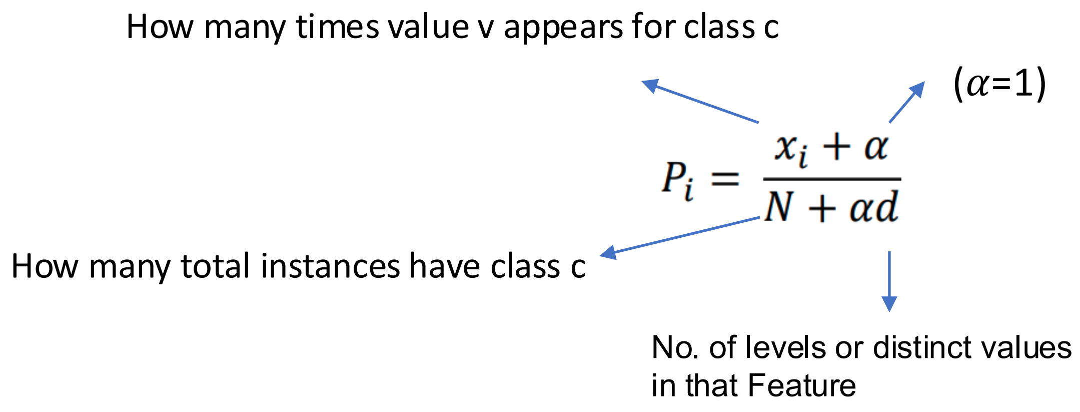
$$P(M \mid Y) = \frac{1+1}{3+3} = \frac{2}{6}$$

$$P(N \mid Y) = \frac{1+1}{3+2} = \frac{2}{5}$$

$$P(T \mid Y) = \frac{0+1}{3+2} = \frac{1}{5}$$

$$= \frac{1}{2}(\frac{2}{6})(\frac{2}{6})(\frac{2}{5})(\frac{1}{5}) = 0.004$$

Instance G will be classified as N.



$$P_i = \frac{x_i + \alpha}{N + \alpha d}$$

How many times value v appears for class c

(α=1)

How many total instances have class c

No. of levels or distinct values in that Feature

ID	Outlook	Temp	Humid	Wind	PLAY
A	S	H	N	F	N
B	S	H	H	T	N
C	O	H	H	F	Y
D	R	M	H	F	Y
E	R	C	N	F	Y
F	R	C	N	T	N

ID	Outlook	Temp	Humid	Wind	PLAY
G	O	M	N	T	?
H	S	M	H	F	?

Instance H

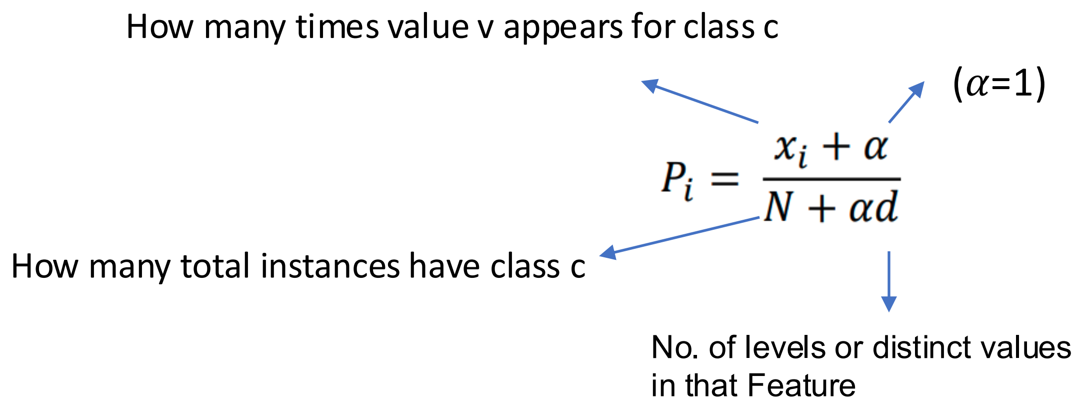
$$N: P(N) \times P(Outlook=S|N) \times P(Temp=M|N) \times P(Humid=H|N) \times P(Wind=F|N)$$

$$= \frac{1}{2}(\frac{3}{6})(\frac{1}{6})(\frac{2}{5})(\frac{2}{5}) = 0.007$$

$$Y: P(Y) \times P(Outlook=S|Y) \times P(Temp=M|Y) \times P(Humid=H|Y) \times P(Wind=F|Y)$$

$$= \frac{1}{2}(\frac{1}{6})(\frac{2}{6})(\frac{3}{5})(\frac{4}{5}) = 0.013$$

Instance H will be classified as Y.



$$P_i = \frac{x_i + \alpha}{N + \alpha d}$$

How many times value v appears for class c

(α=1)

How many total instances have class c

No. of levels or distinct values in that Feature

ID	Outlook	Temp	Humid	Wind	PLAY
A	S	H	N	F	N
B	S	H	H	T	N
C	O	H	H	F	Y
D	R	M	H	F	Y
E	R	C	N	F	Y
F	R	C	N	T	N

ID	Outlook	Temp	Humid	Wind	PLAY
G	O	M	N	T	N
H	S	M	H	F	?

In step 3, we add confidently-classified instances to the training dataset

Instance G

$$N{:}P(N)\times P(Outlook{=}O\,|\,N)\times P(Temp{=}M\,|\,N)\times P(Humid{=}N\,|\,N)\times P(Wind{=}T\,|\,N)$$

$$=\frac{1}{2}(\frac{1}{6})(\frac{1}{6})(\frac{3}{5})(\frac{3}{5})=0.005$$

Instance H

$$Y{:}P(Y)\times P(Outlook{=}S\,|\,Y)\times P(Temp{=}M\,|\,Y)\times P(Humid{=}H\,|\,Y)\times P(Wind{=}F\,|\,Y)$$

$$=\frac{1}{2}(\frac{1}{6})(\frac{2}{6})(\frac{3}{5})(\frac{4}{5})=0.013$$

We could define a "confident" classification as a classification with posterior probability >0.01

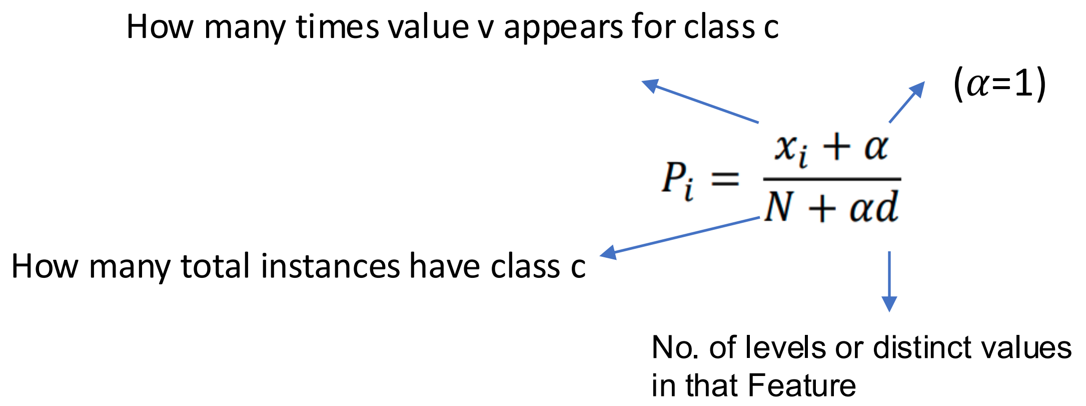
Using this definition, instance H has been confidently classified and should be added to the training dataset with the label Y.

In step 4, we would then retrain the naive Bayes model using the new training dataset (A-F + H) and continue trying to label the unlabelled instances (G). With the new model, instance G will be classified as follows:

$$N : \frac{3}{7}\left(\frac{1}{6}\right)\left(\frac{1}{6}\right)\left(\frac{3}{5}\right)\left(\frac{3}{5}\right) = 0.0042$$

$$Y : \frac{4}{7}\left(\frac{2}{7}\right)\left(\frac{3}{7}\right)\left(\frac{2}{6}\right)\left(\frac{1}{6}\right) = 0.0038$$

Instance G will be classified as N.



How many times value v appears for class c

(α=1)

$$P_i = \frac{x_i + \alpha}{N + \alpha d}$$

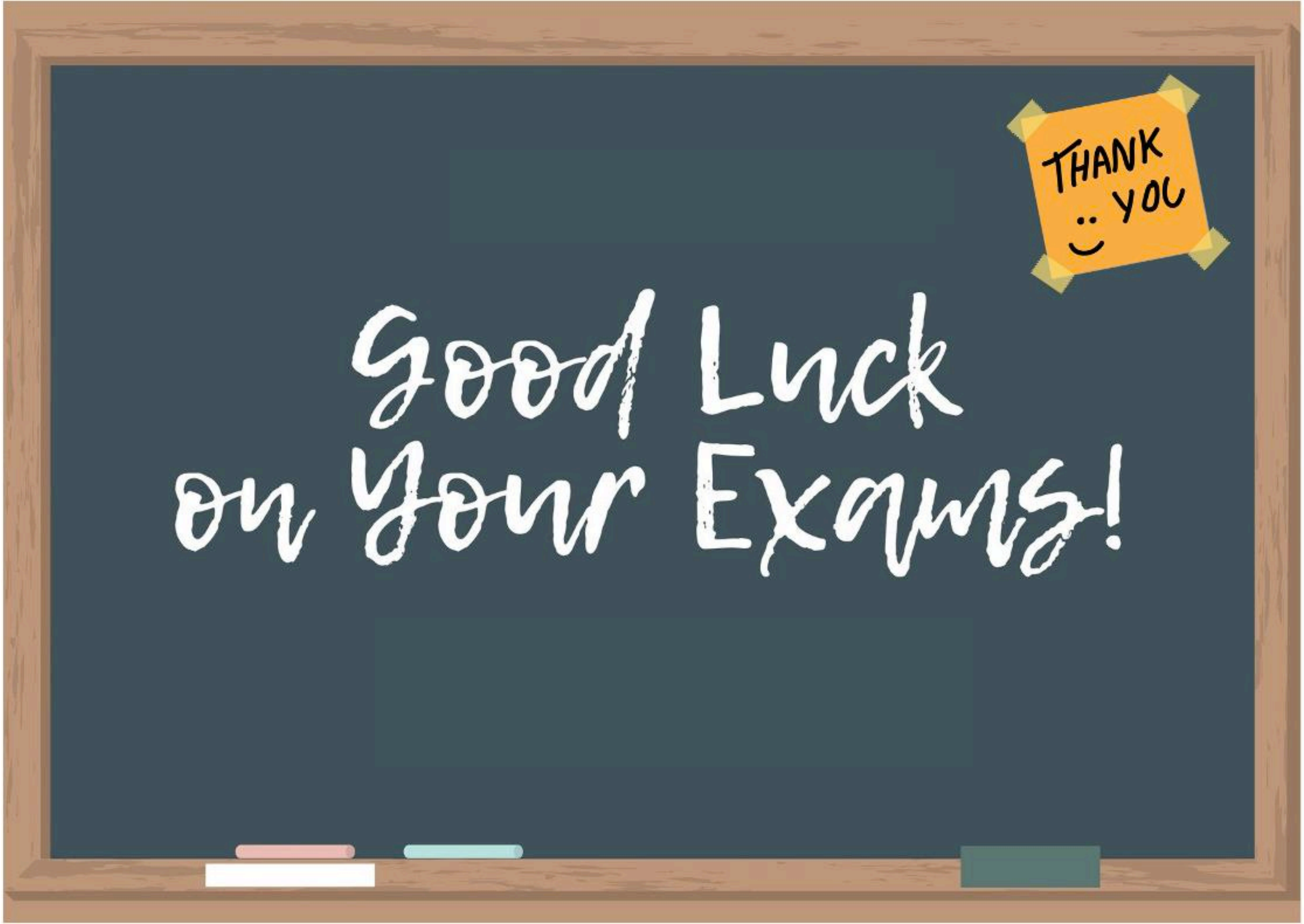
How many total instances have class c

No. of levels or distinct values in that Feature

ID	Outlook	Temp	Humid	Wind	PLAY
A	S	H	N	F	N
B	S	H	H	T	N
C	O	H	H	F	Y
D	R	M	H	F	Y
E	R	C	N	F	Y
F	R	C	N	T	N
H	S	M	H	F	Y

ID	Outlook	Temp	Humid	Wind	PLAY
G	O	M	N	T	?

Given our threshold for a "confident" classification is 0.01, instance G still cannot be confidently classified, so the self-training algorithm will terminate at this iteration.



Good Luck on Your Exams!