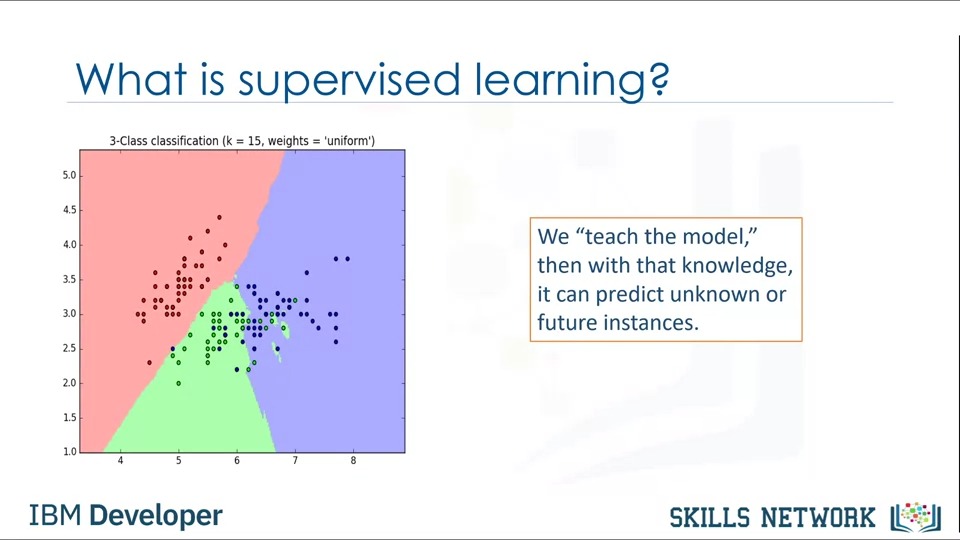
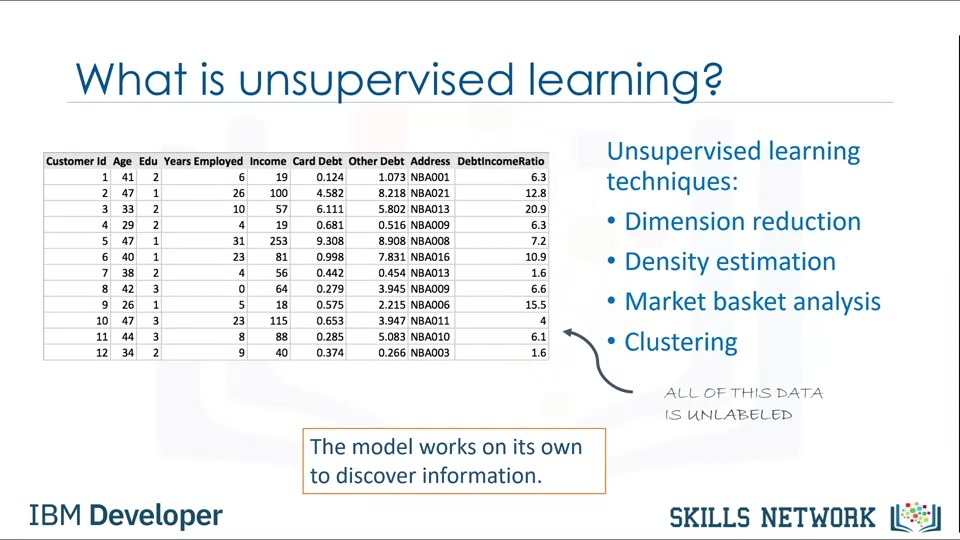
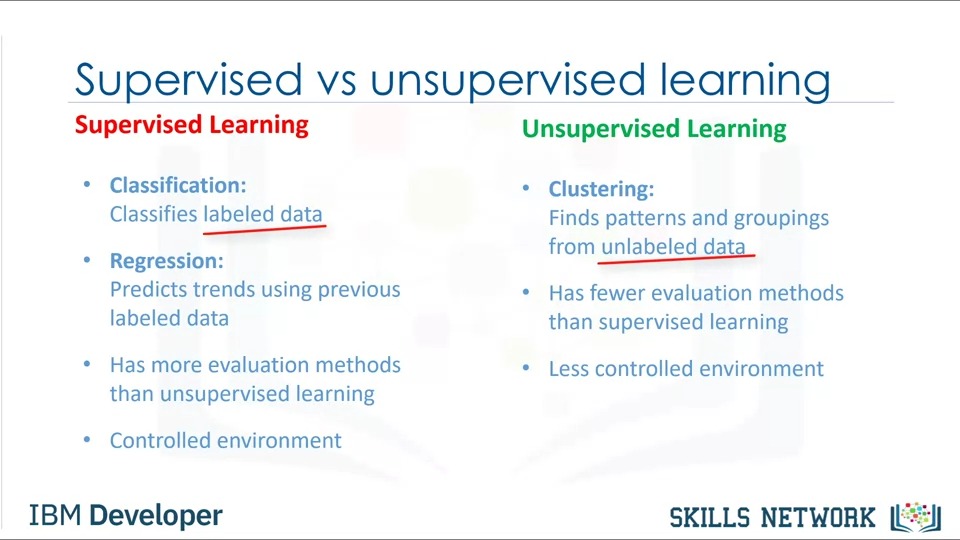
# **Supervised Vs unsupervised learning**

So, to recap, the biggest difference between supervised and unsupervised learning is that supervised learning deals with labeled data while unsupervised learning deals with unlabeled data.

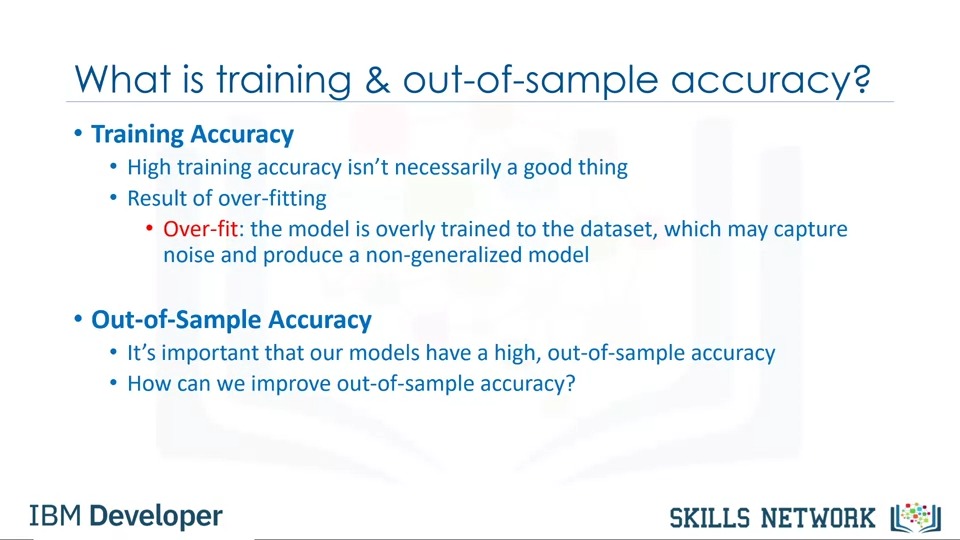




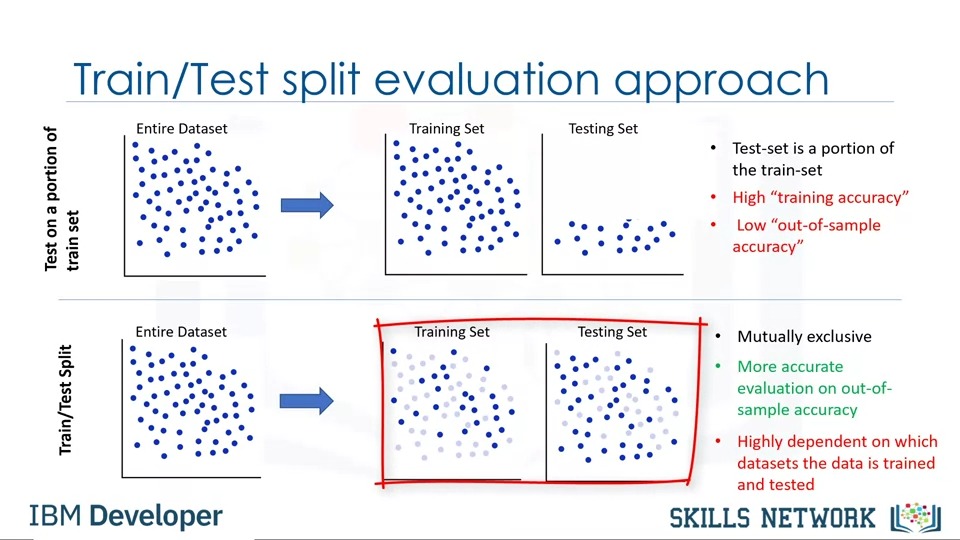


# **Regression Models**

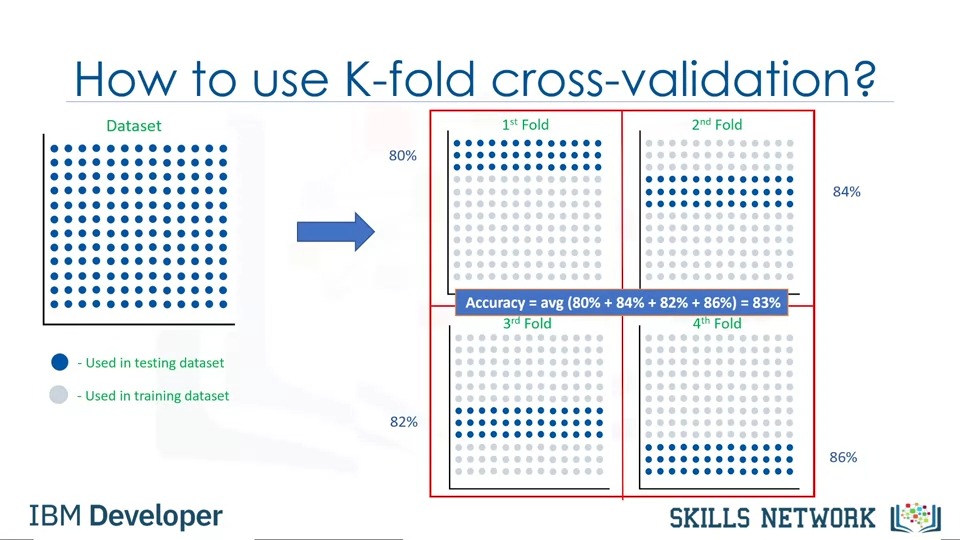
## **Model Evaluation in Regression Models**



So, how can we improve out-of-sample accuracy? One way is to use another evaluation approach called train/test split. The issue with train/test split is that it's highly dependent on the datasets on which the data was trained and tested.

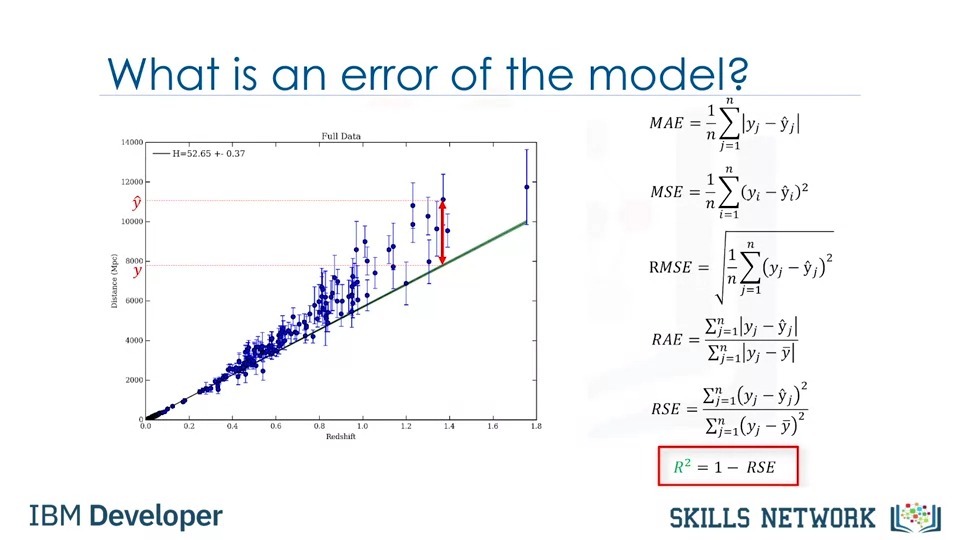


K-fold cross-validation leads to a more consistent out-of-sample accuracy.

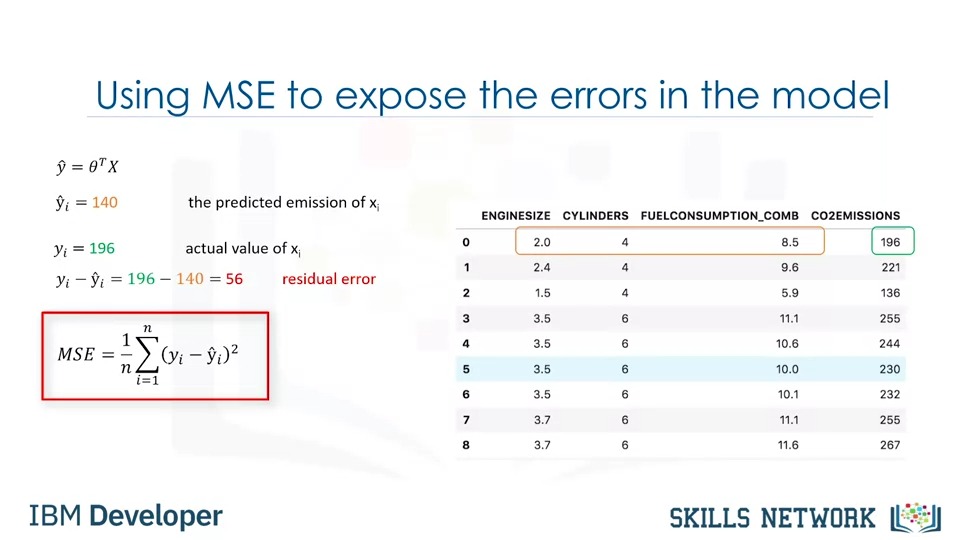


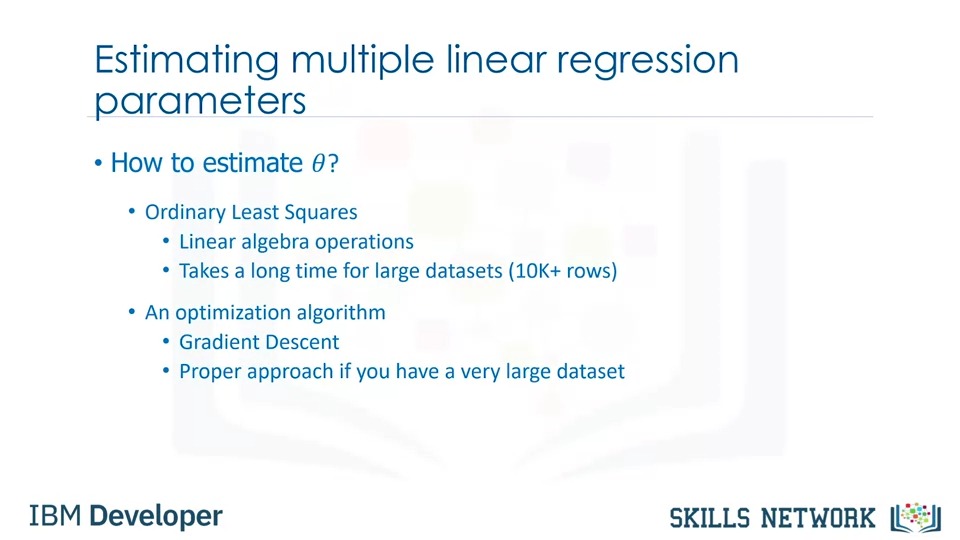
## **Evaluation Metrics in Regression Models**

The higher the R-squared, the better the model fits your data.

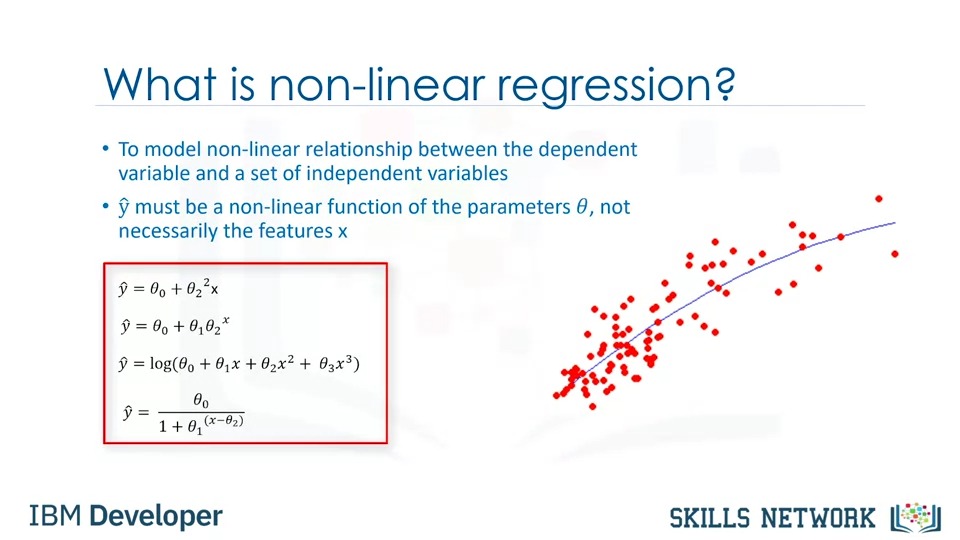


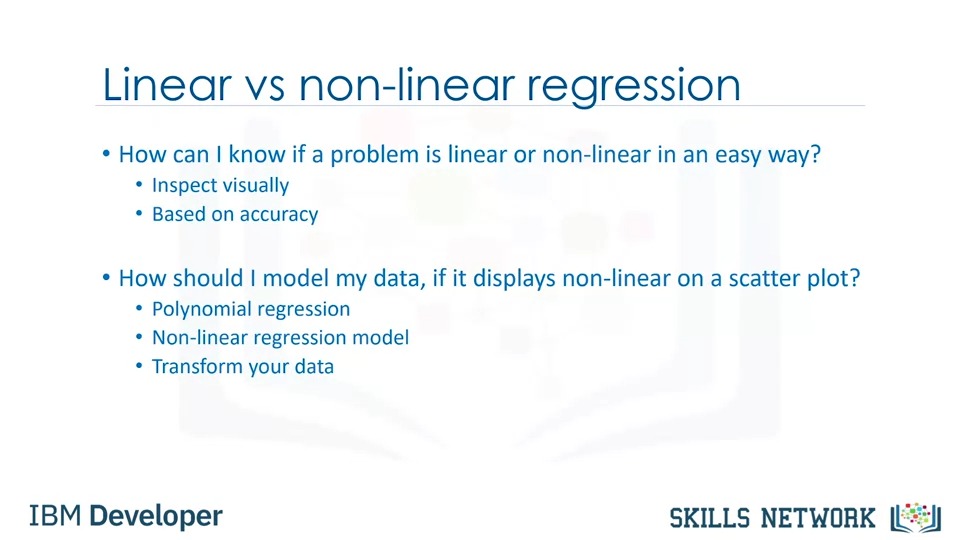
## **Multiple Linear Regression**



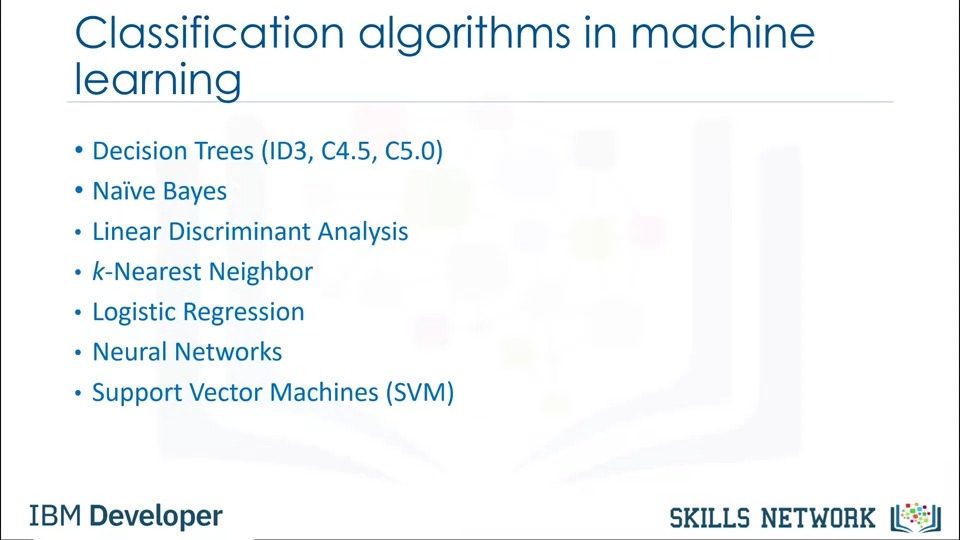


## **Non-linear Regression**





# **Classification**

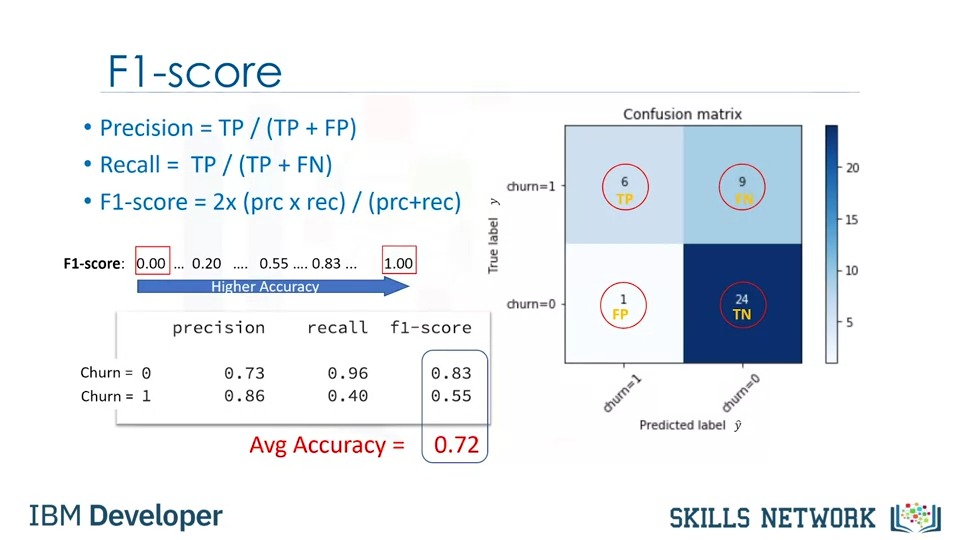


## **Evaluation Metrics in Classification**

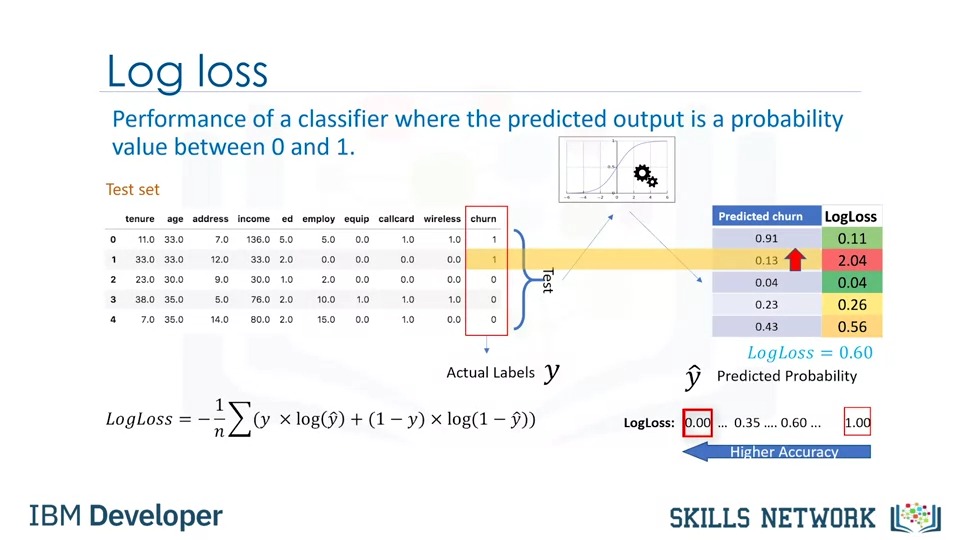
If the entire set of predicted labels for a sample strictly matches with the true set of labels, then the subset accuracy is 1.0; otherwise it is 0.0.



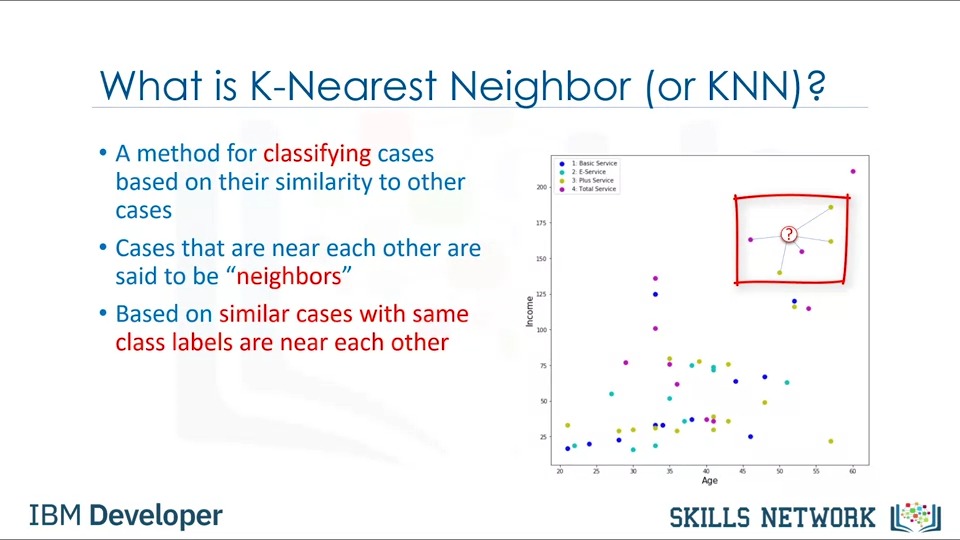
So, the classifier with higher F1-score has better accuracy. Please notice that both Jaccard and F1-score can be used for multi-class classifiers as well.

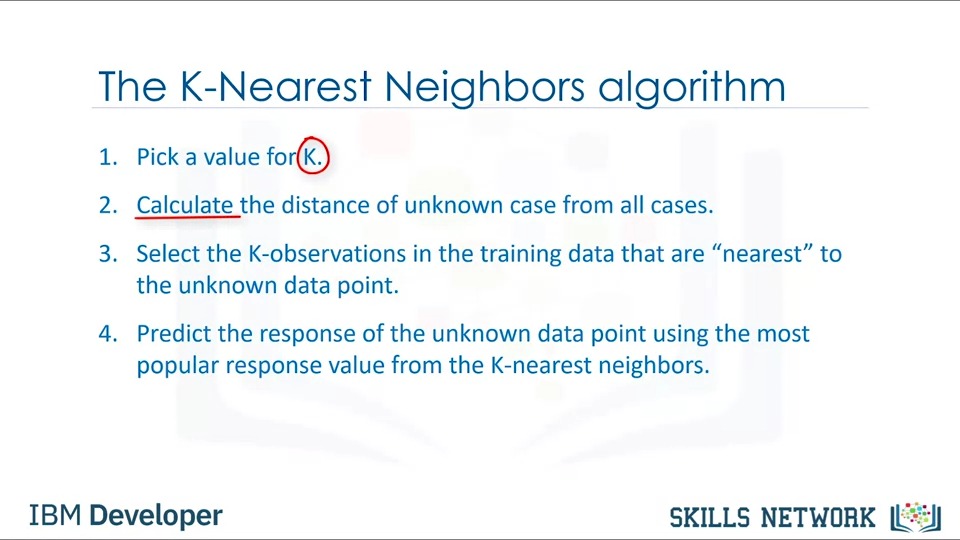


So, the classifier with lower log loss has better accuracy.

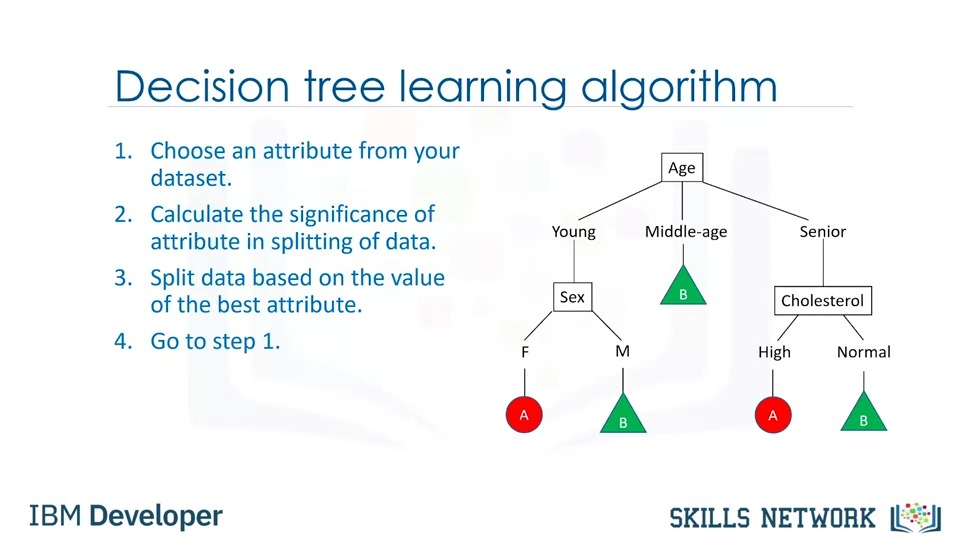


## **Classification: K-Nearest Neighbours (KNN)**

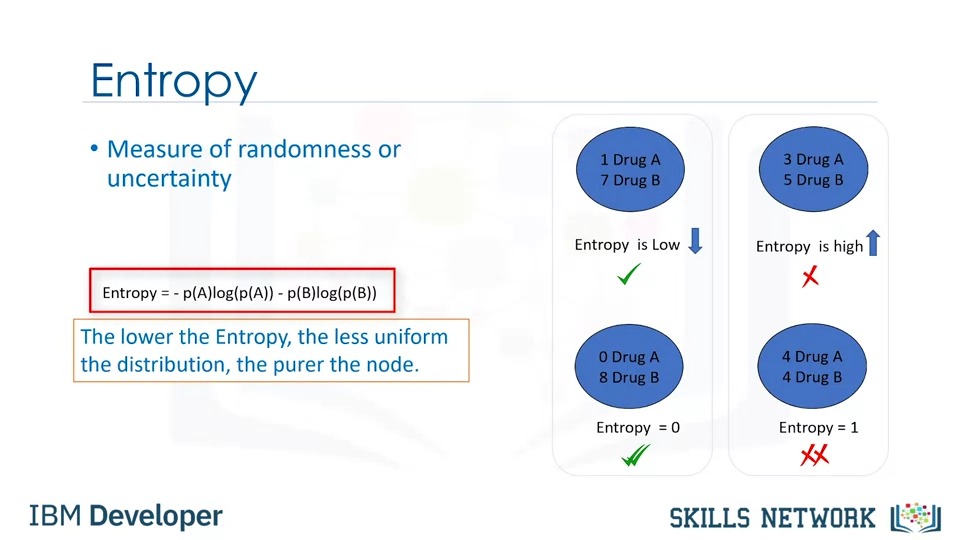




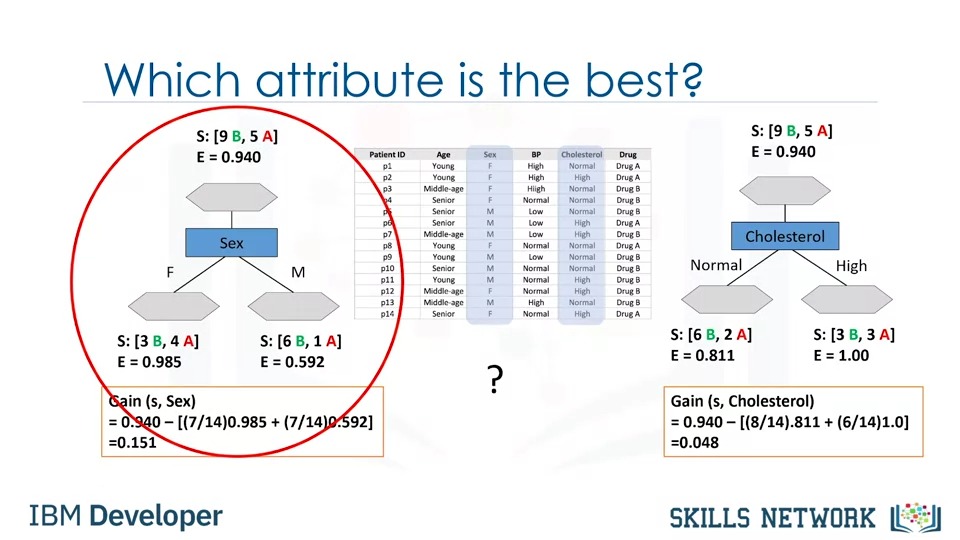
## **Classification: Decision Trees**



You can easily calculate the entropy of a node using the frequency table of the attribute through the entropy formula where P is for the proportion or ratio of a category, such as drug A or B.

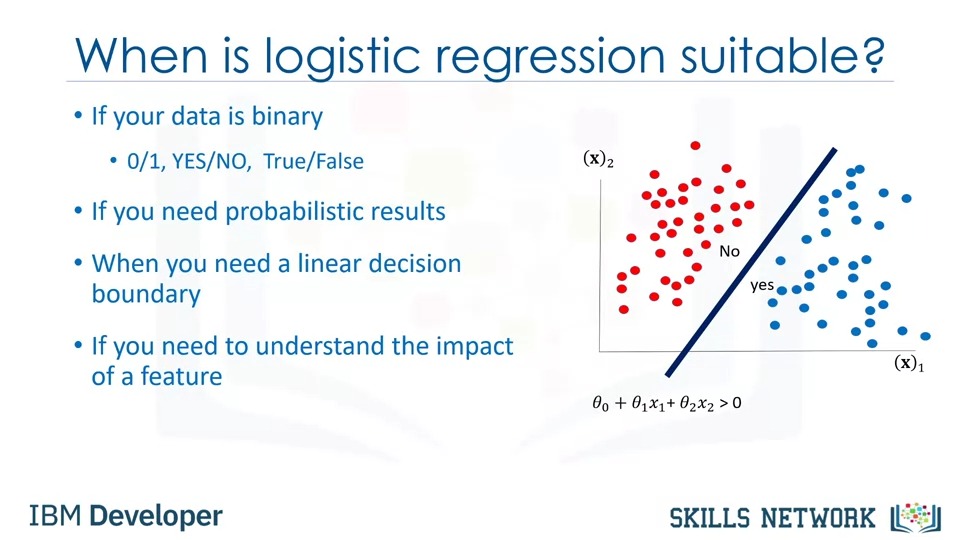


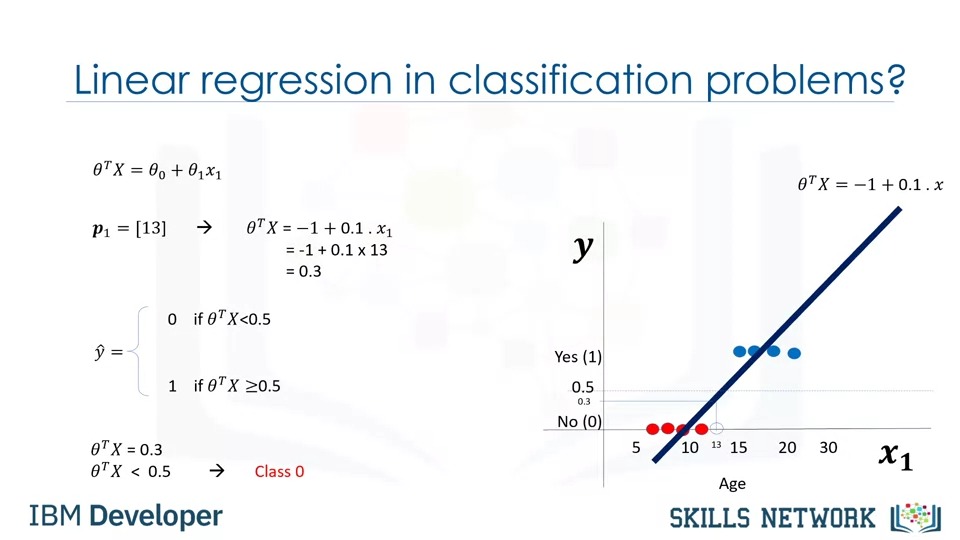
Now, the question is, which attribute is more suitable? Well, as mentioned, the tree with the higher information gained after splitting, this means the sex attribute.

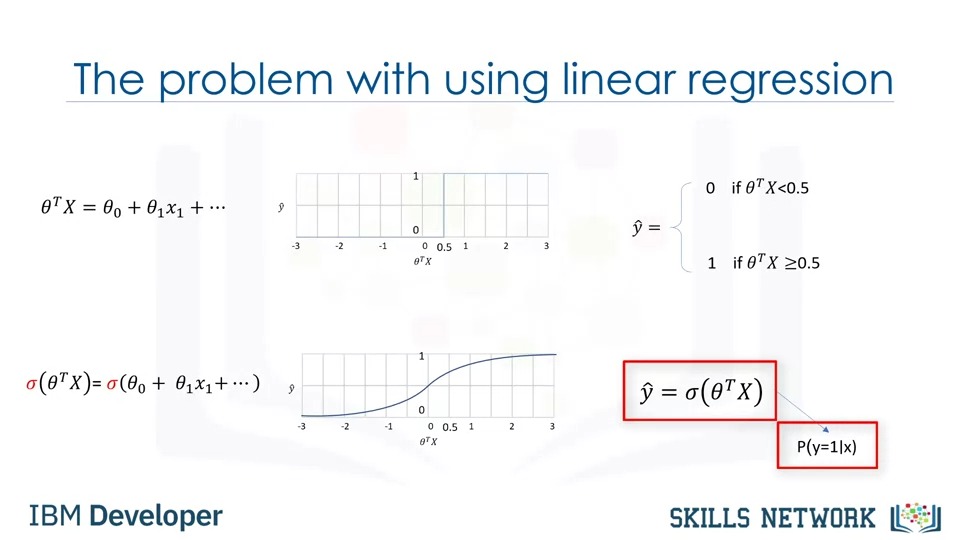


## **Classification: Logistic Regression**

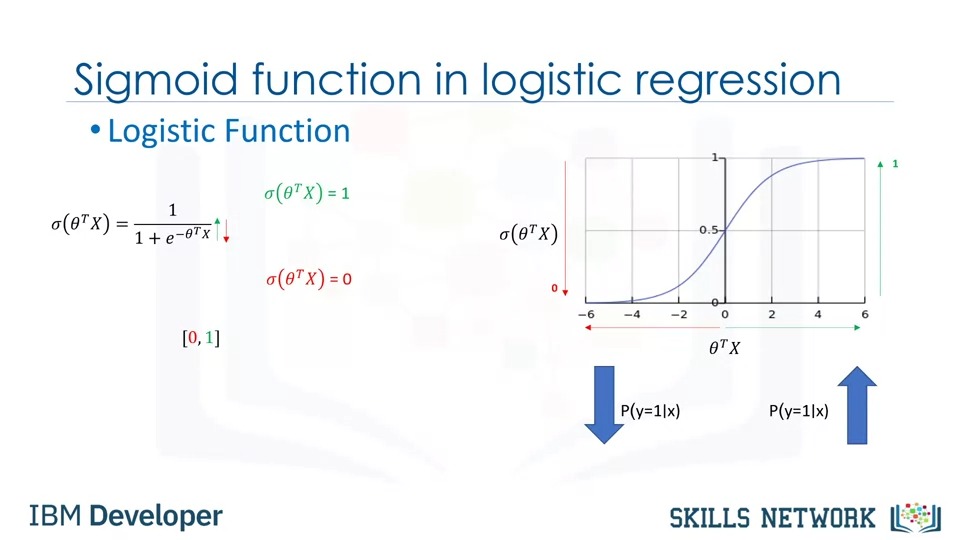




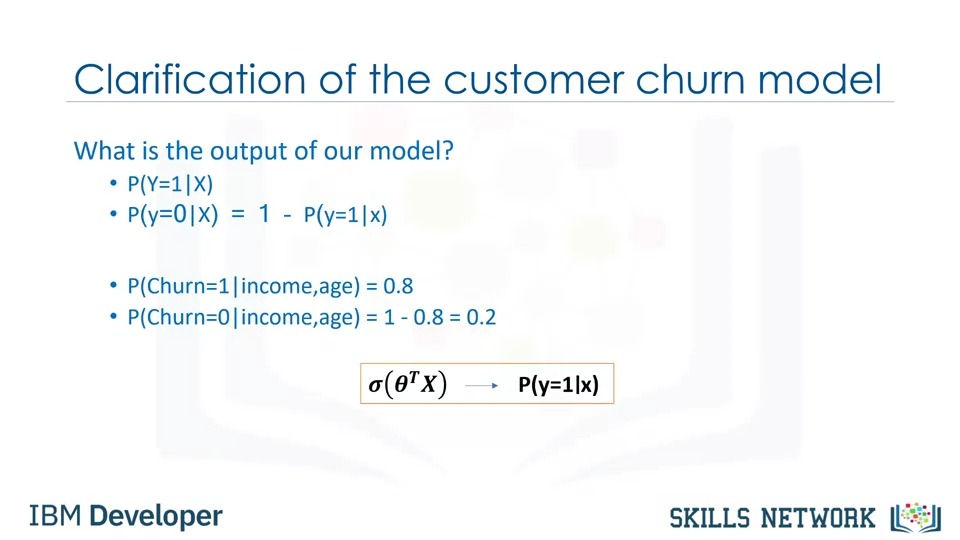
Also, there are some other issues which verify that linear regression is not the proper method for classification problems.

Now, our model is sigmoid of Theta transpose x, which represents the probability that the output is 1 given x.

In contrast, when the sigmoid value is closer to 0, the probability of y equals 1 given x is very small.

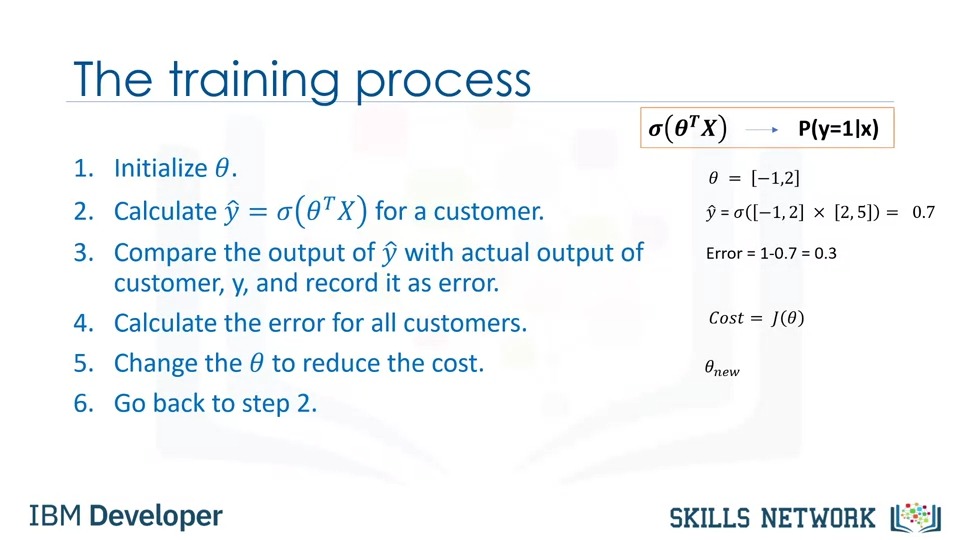


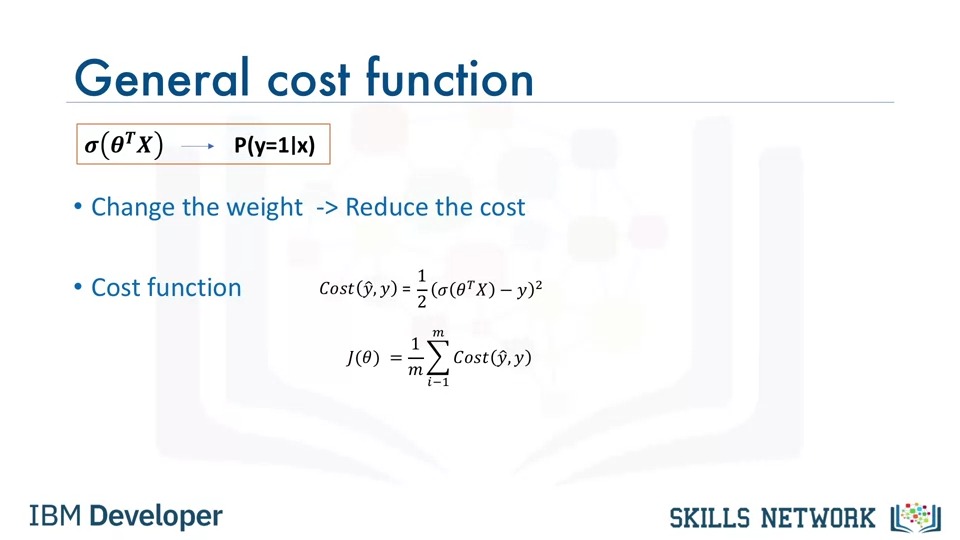
So, now our job is to train the model to set its parameter values in such a way that our model is a good estimate of probability of y equals 1 given x.



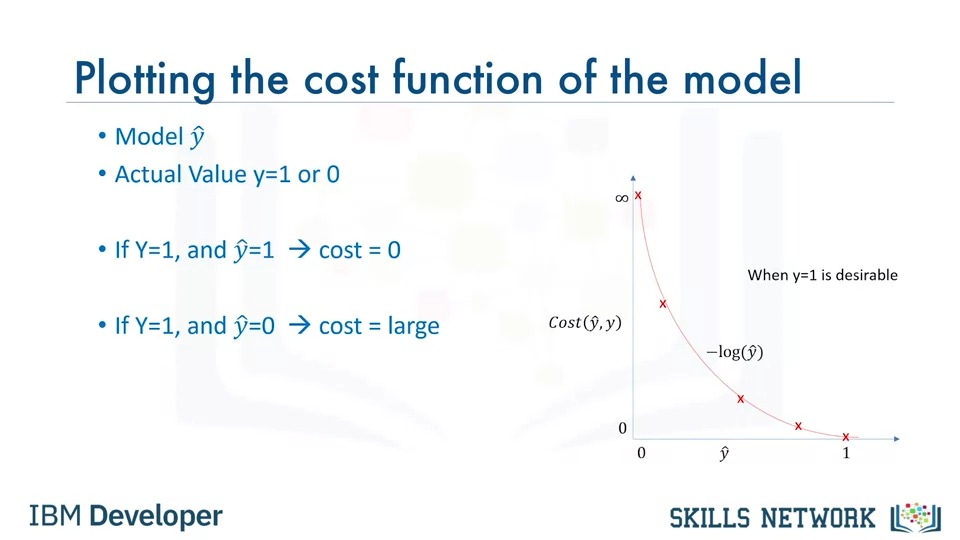
### **Training Logistic Regression Model**

First, how can we change the values of Theta so that the cost is reduced across iterations? Second, when should we stop the iterations? There are different ways to change the values of Theta, but one of the most popular ways is gradient descent.





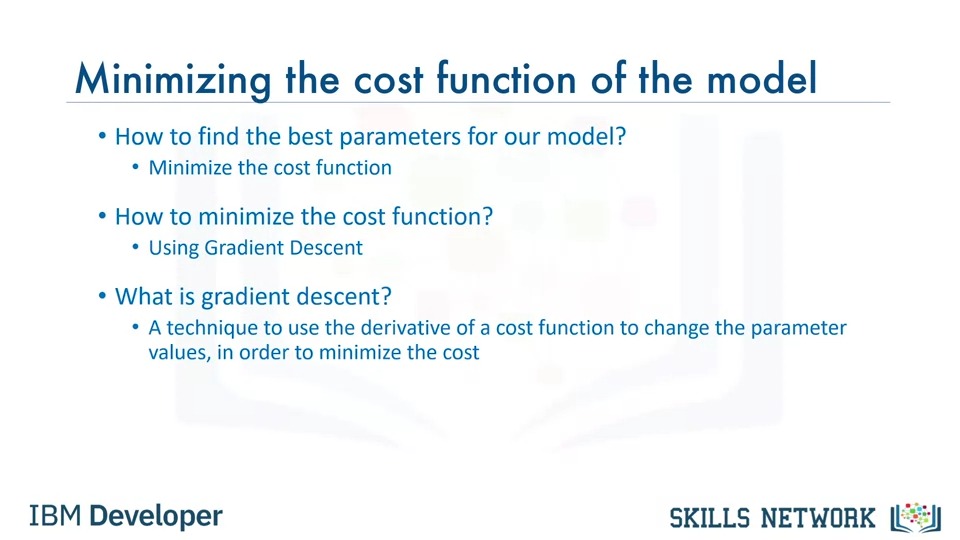
So, what is the solution? Well we should find another cost function instead, one which has the same behavior but is easier to find its minimum point. It means if the actual value is one and the model also predicts one, the minus log function returns zero cost.



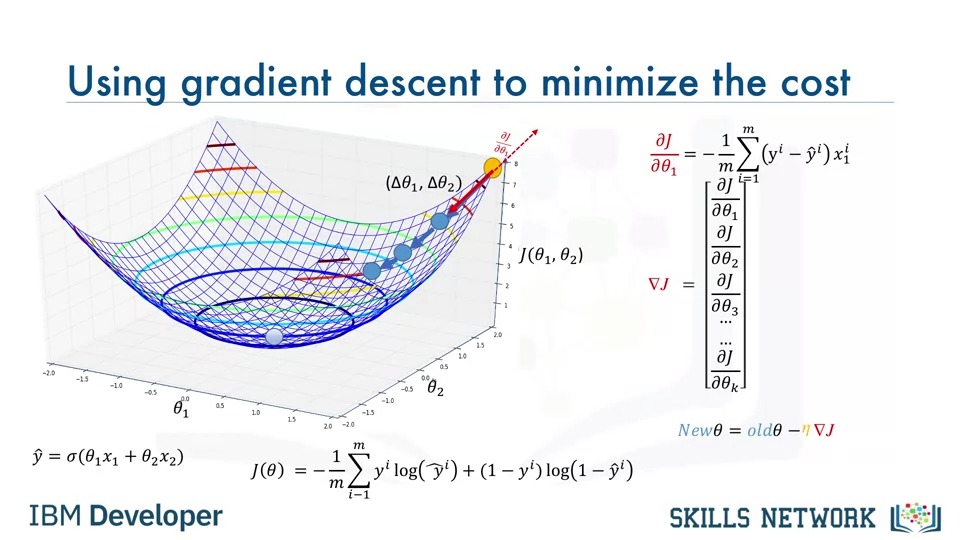
Now, we can easily use this function to find the parameters of our model in such a way as to minimize the cost.



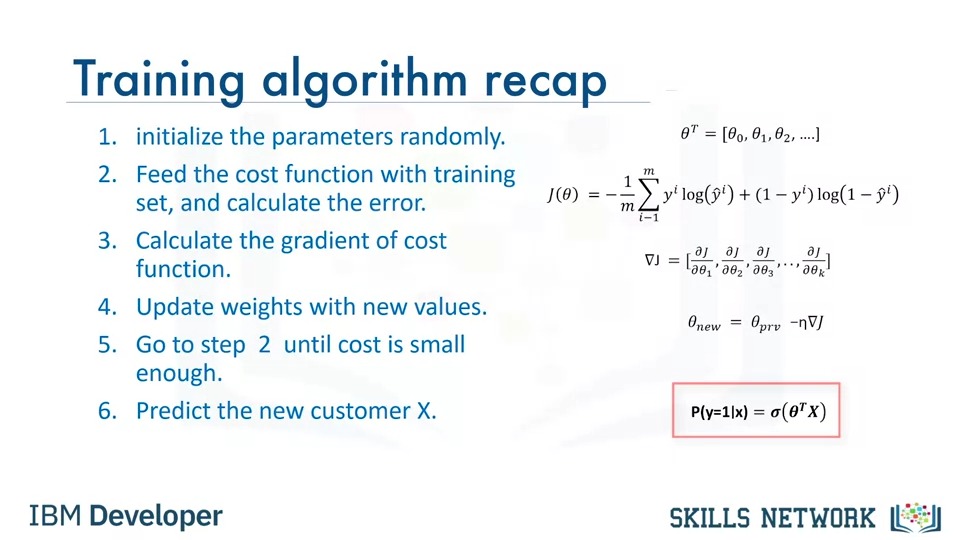
Specifically in our case gradient descent is a technique to use the derivative of a cost function to change the parameter values to minimize the cost or error.



Notice that it's an iterative operation and in each iteration we update the parameters and minimize the cost until the algorithm converge is on an acceptable minimum.

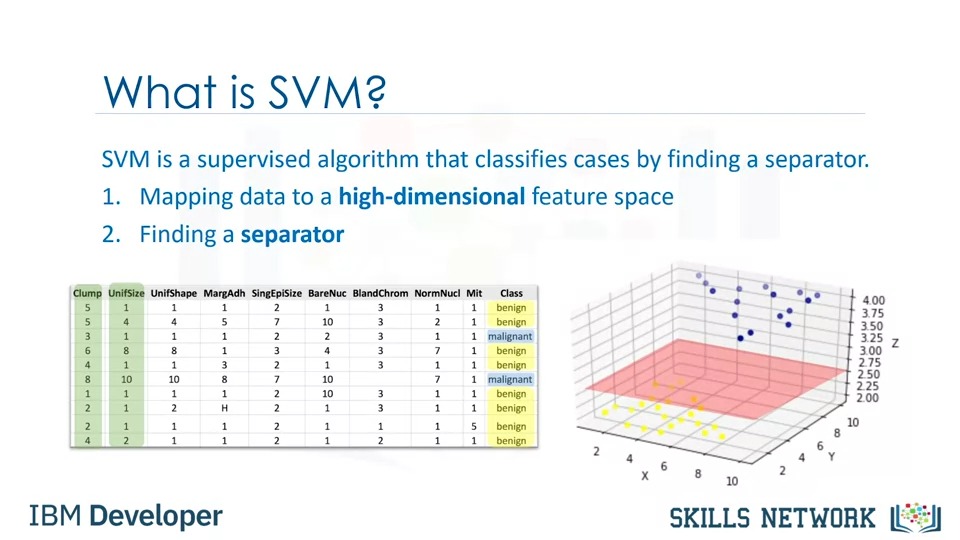


This means the model is ready and we can use it to predict the probability of a customer staying or leaving.

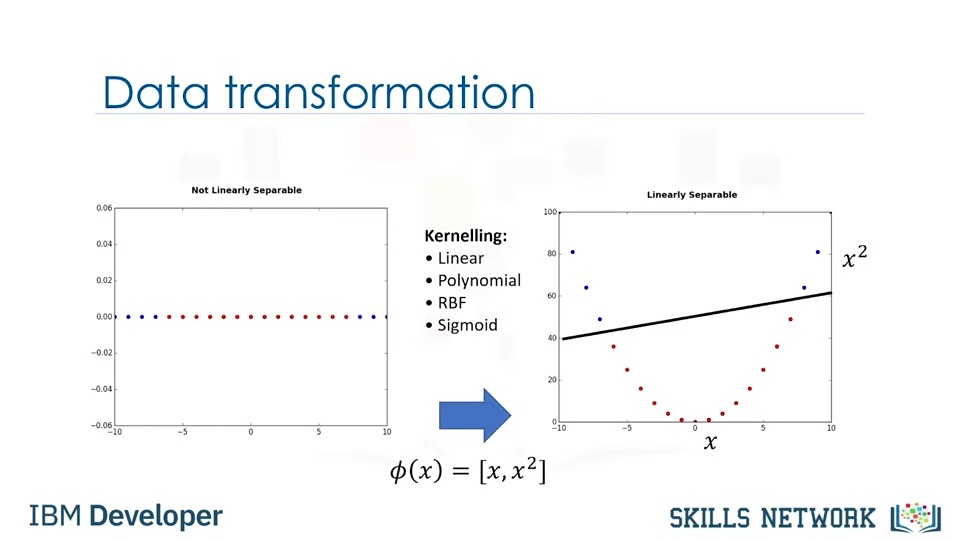


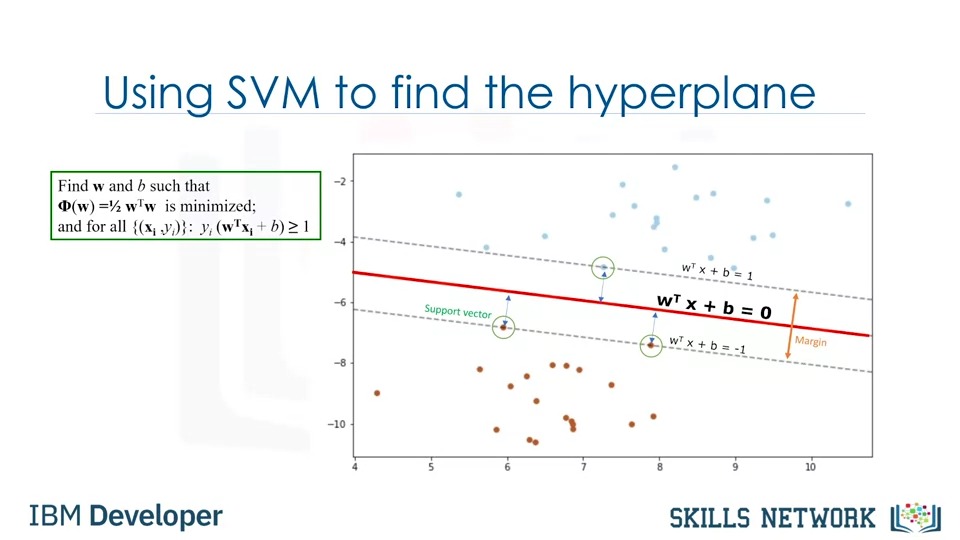
## **Classification: Support Vector Machine (SVM)**

First, how do we transfer data in such a way that a separator could be drawn as a hyperplane? And two, how can we find the best or optimized hyperplane separator after transformation? Let's first look at transforming data to see how it works.

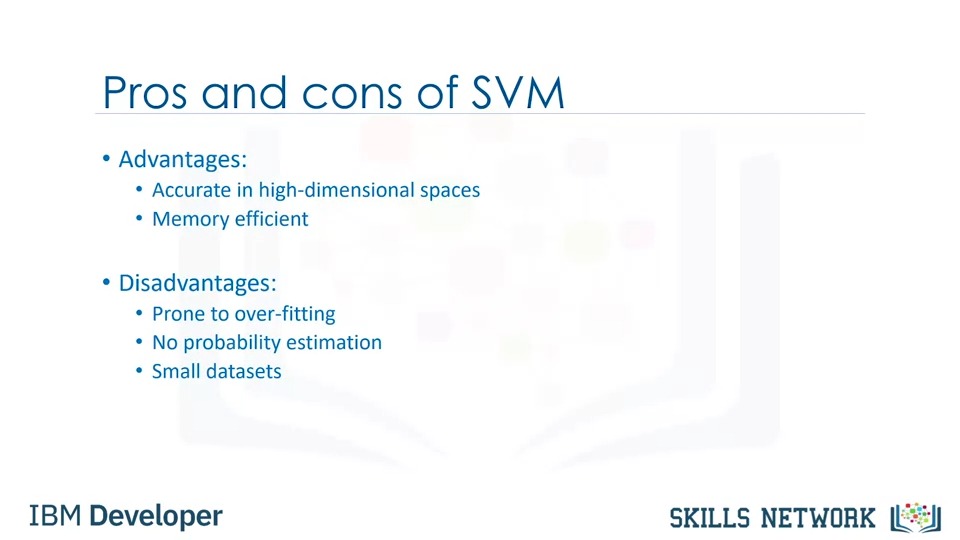


Also, as there's no easy way of knowing which function performs best with any given dataset, we usually choose different functions in turn and compare the results.

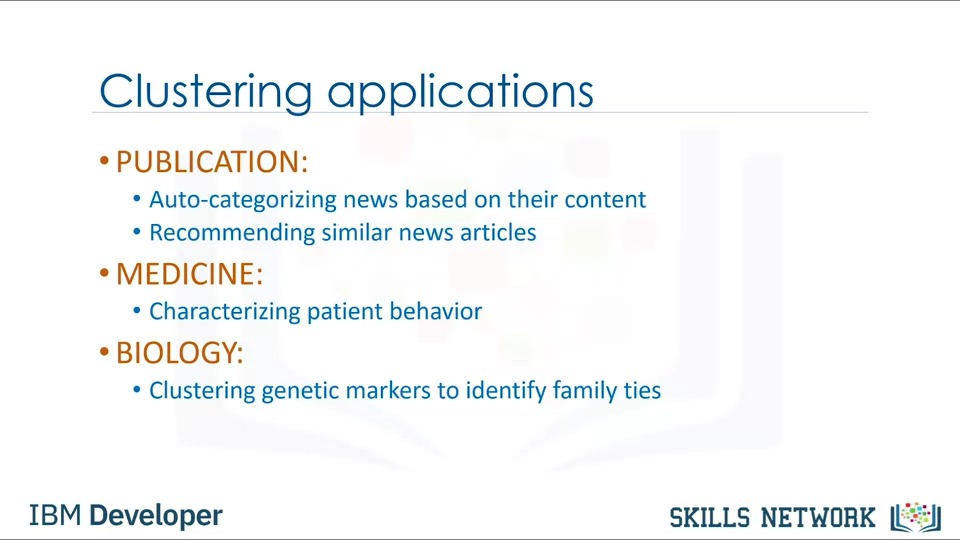


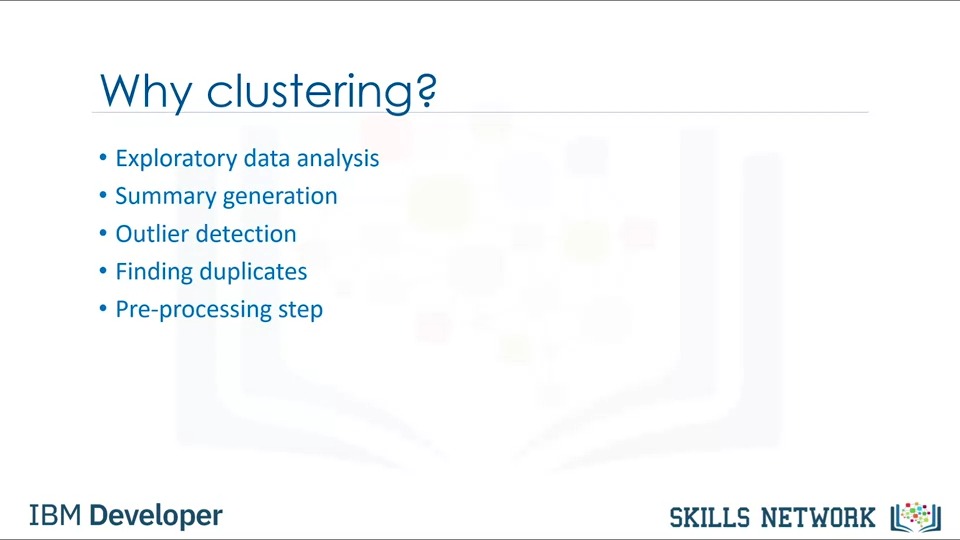


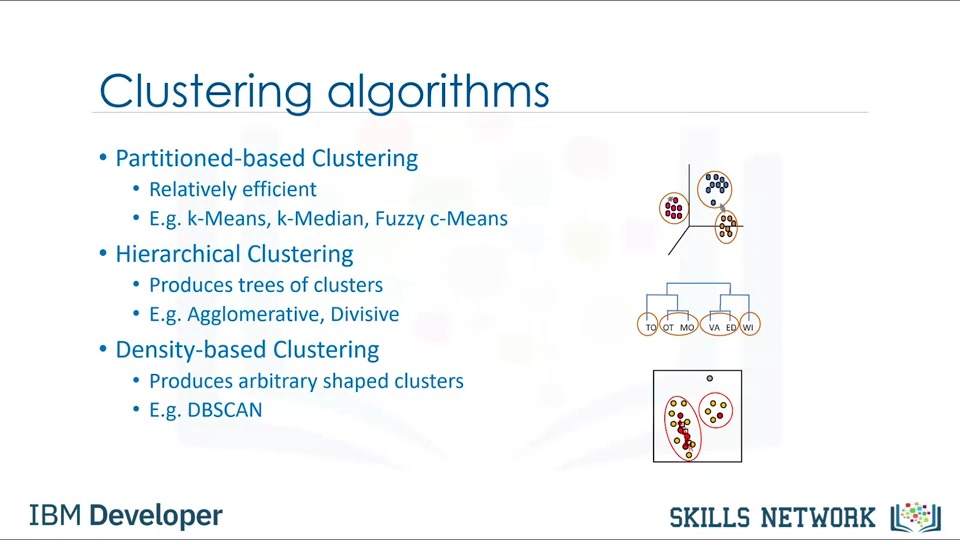




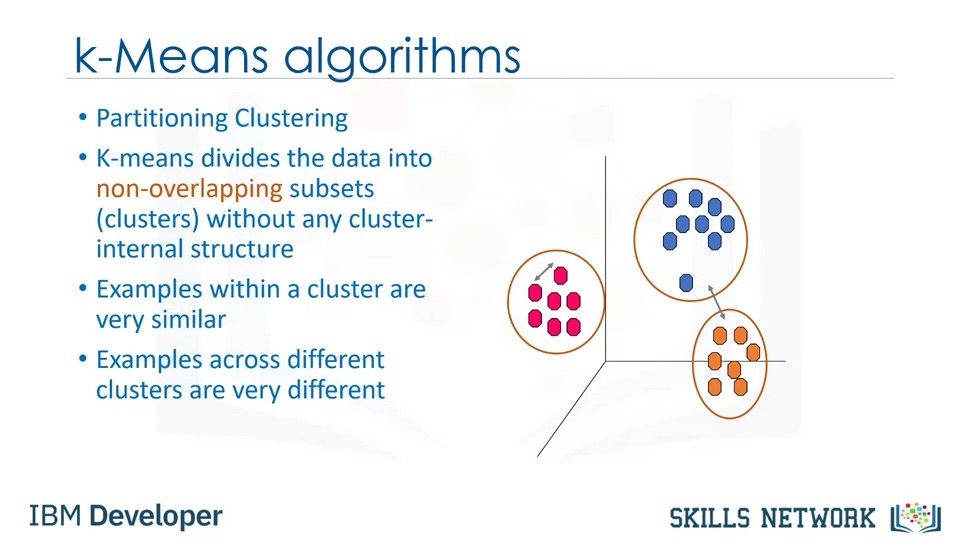
# **Clustering**



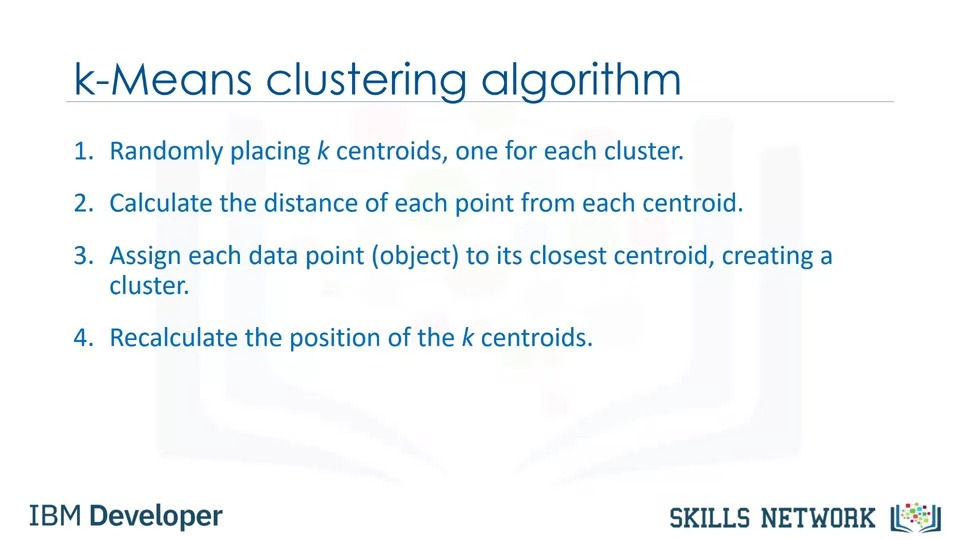




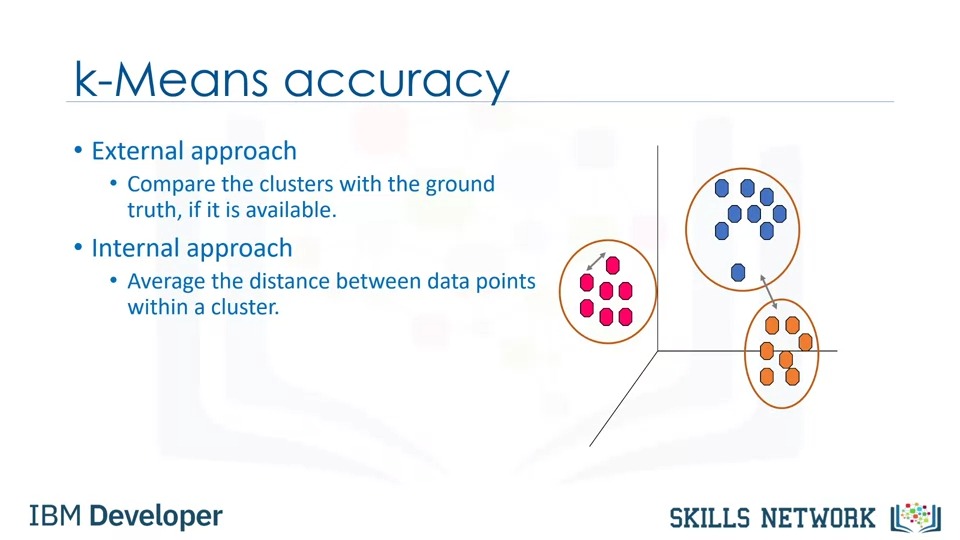
## **Clustering: k-Means Clustering**

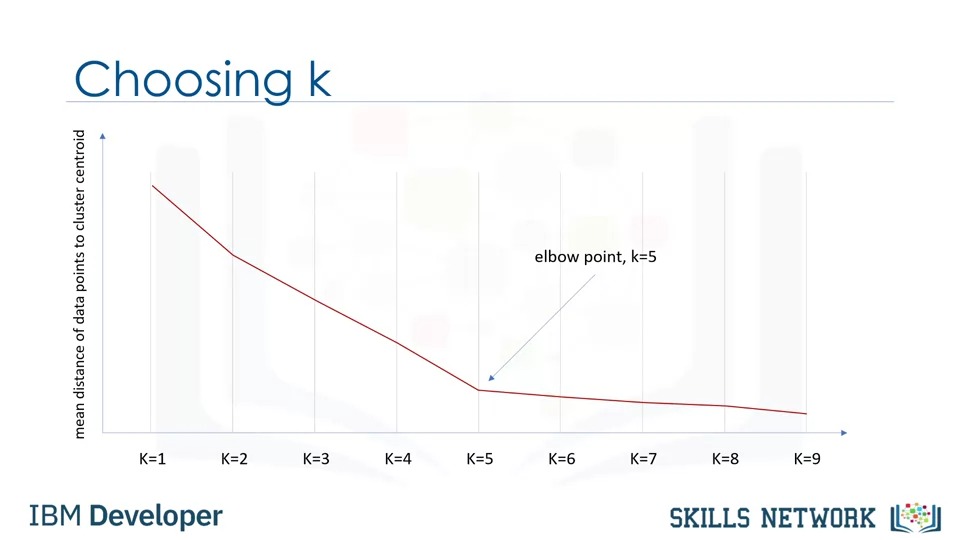


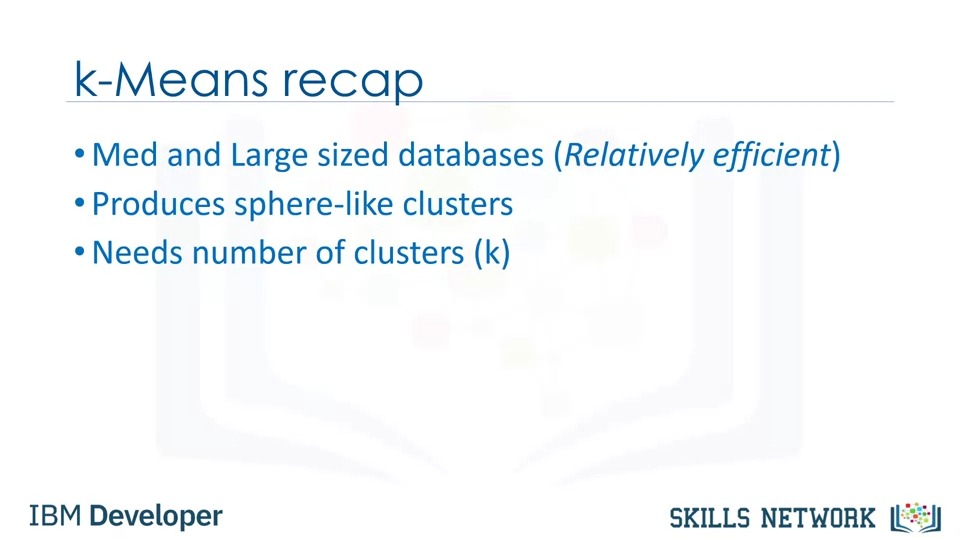
The new centroid position is determined by the mean of all points in the group.



Also, average of the distances of data points from their cluster centroids can be used as a metric of error for the clustering algorithm.

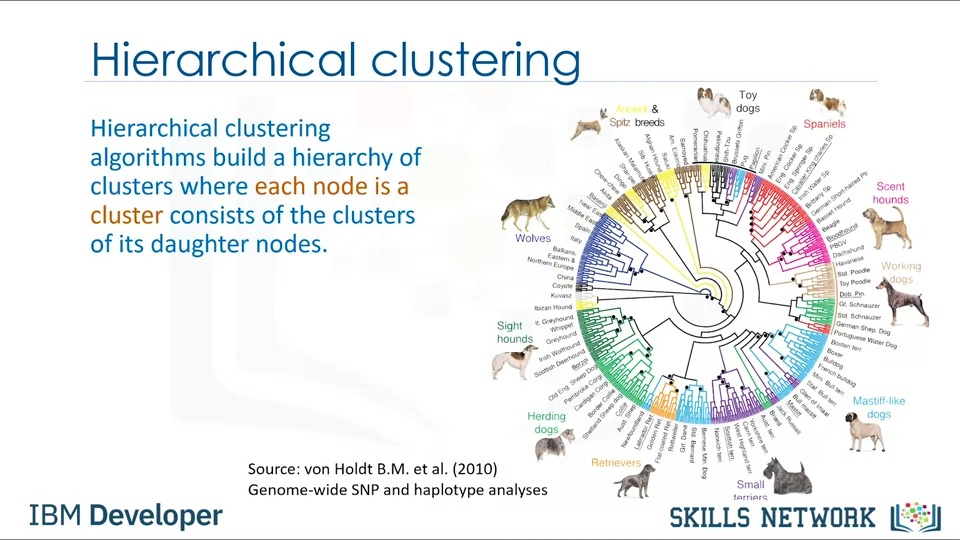




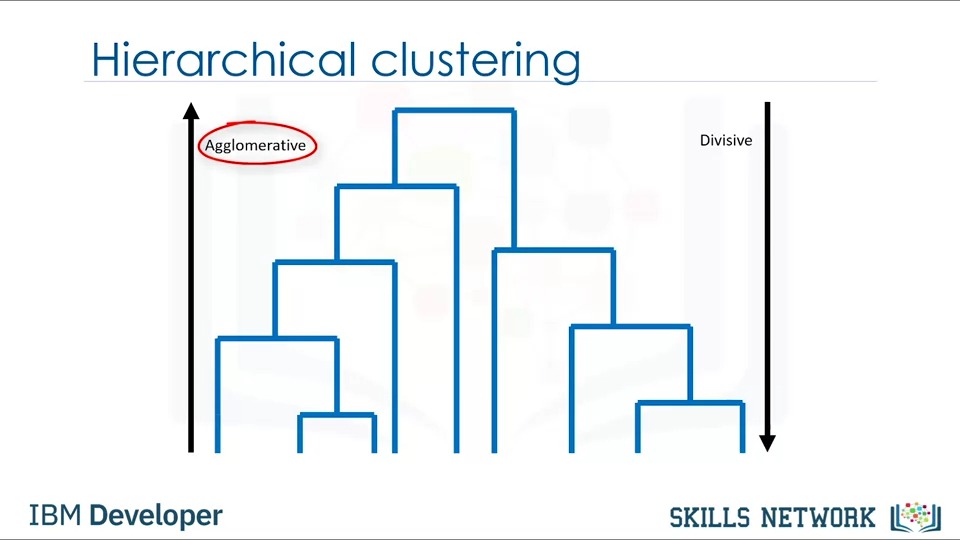


## **Clustering: Hierarchical Clustering**

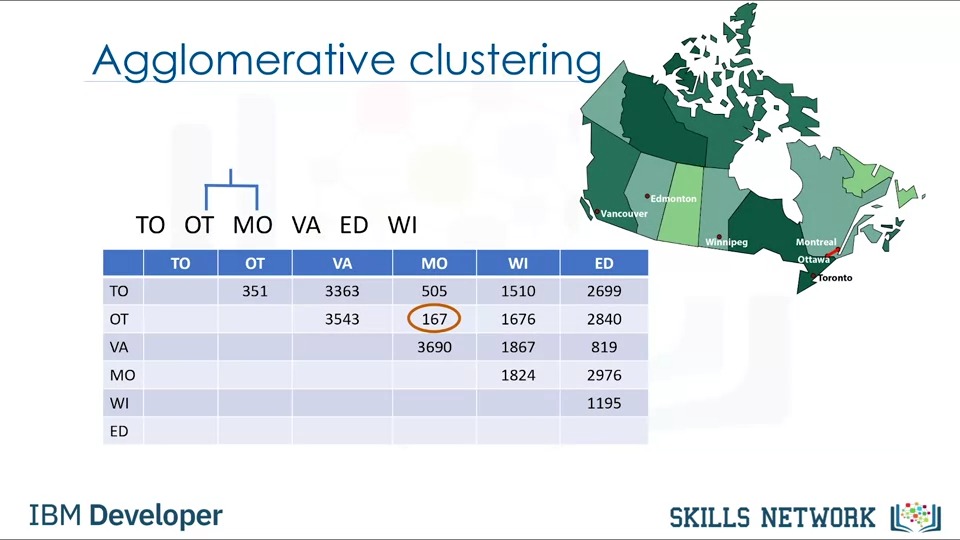
Hierarchical clustering algorithms build a hierarchy of clusters where each node is a cluster consisting of the clusters of its daughter nodes.



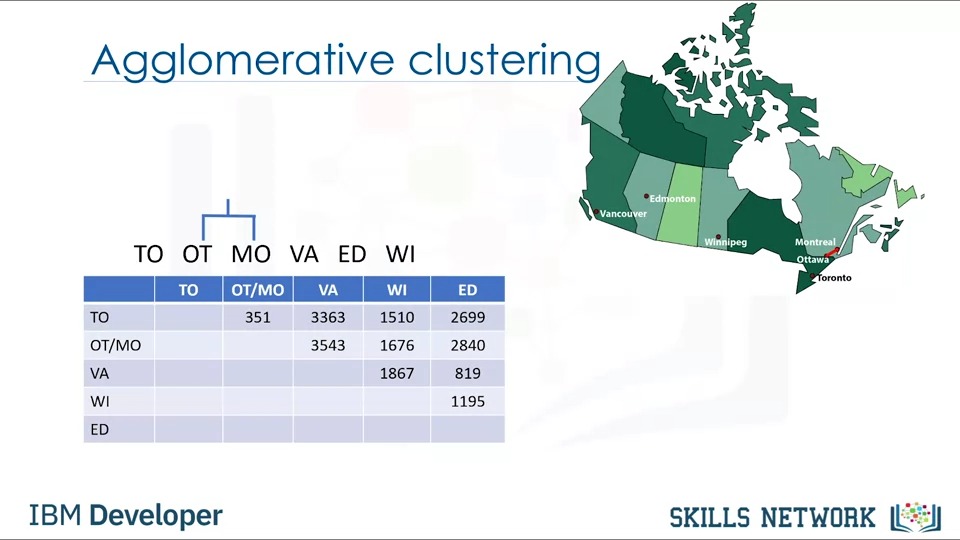
So it is bottom up, where each observation starts in its own cluster and pairs of clusters are merged together as they move up the hierarchy.



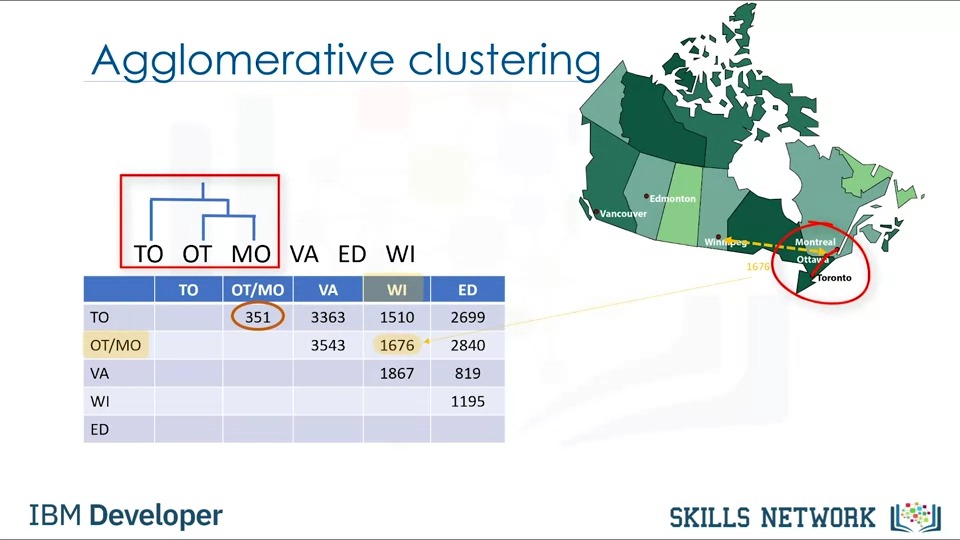
Please notice that we just use a simple one-dimensional distance feature here, but our object can be multidimensional and distance measurement can either be Euclidean, Pearson, average distance or many others depending on data type and domain knowledge.

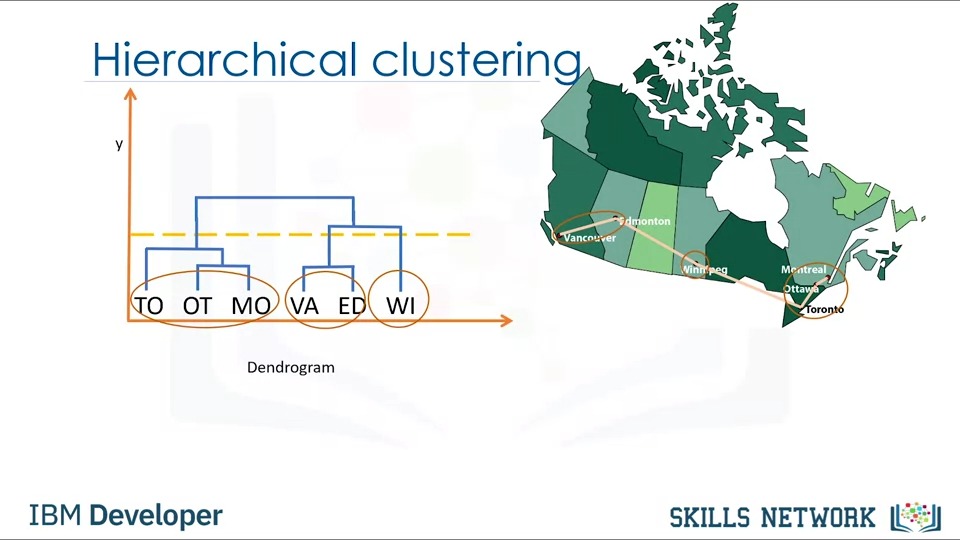


As you can see in the distance matrix, rows and columns related to Montreal and Ottawa cities are merged as the cluster is constructed.



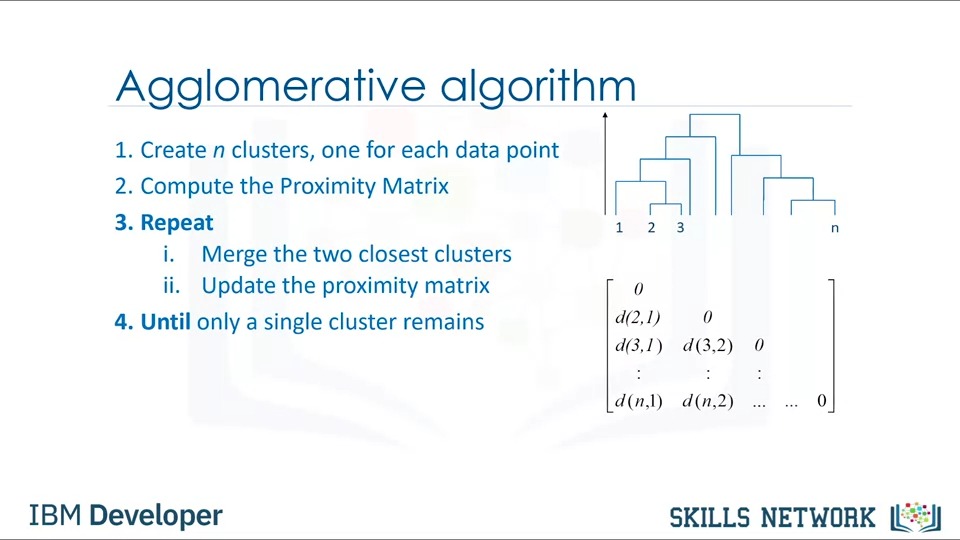
In this case, Ottawa, Montreal, and Toronto are the closest ones which creates another cluster.



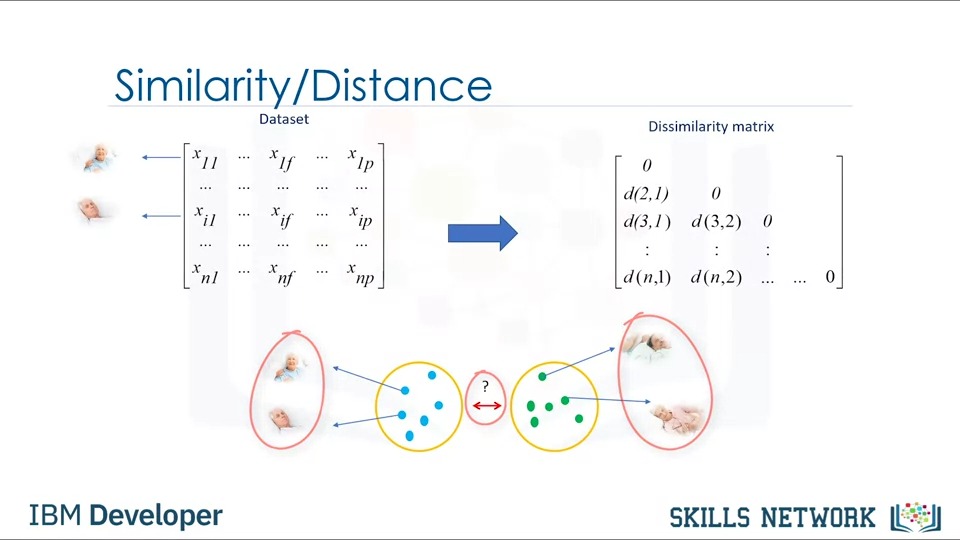


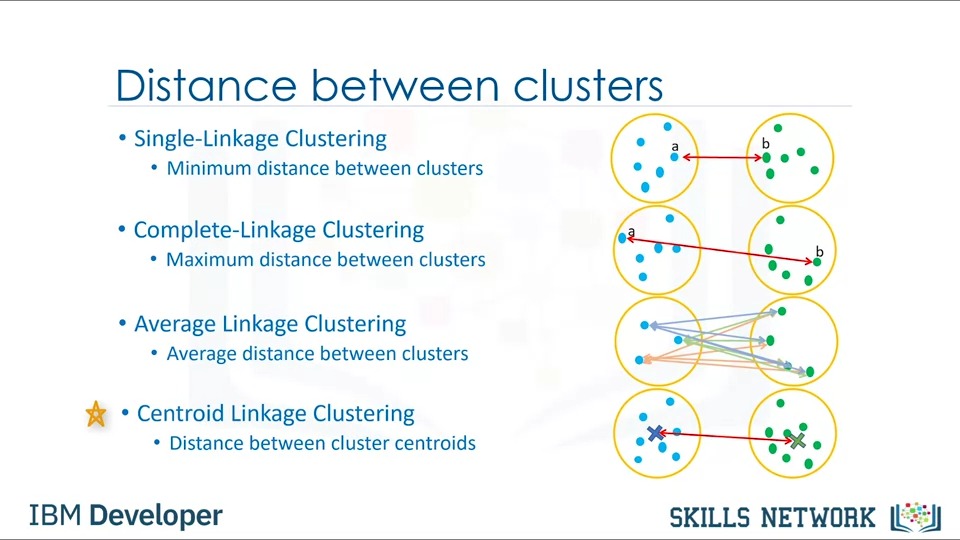
### **Clustering: Compare Hierarchical Clustering with K-Mean**

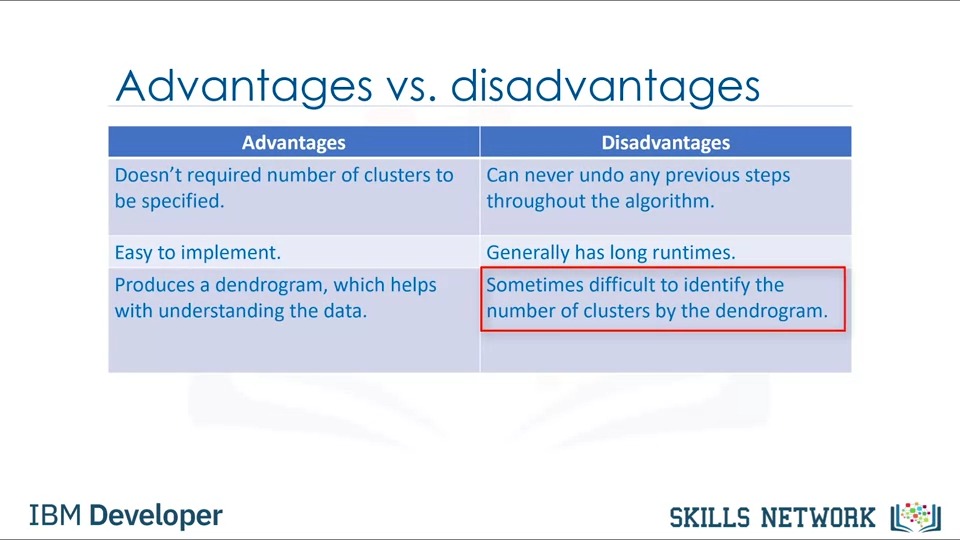
For instance, how do we measure the distances between these clusters and how do we define the nearest among clusters? We also can ask, which points do we use? First, let's see how to calculate the distance between two clusters with one point each.

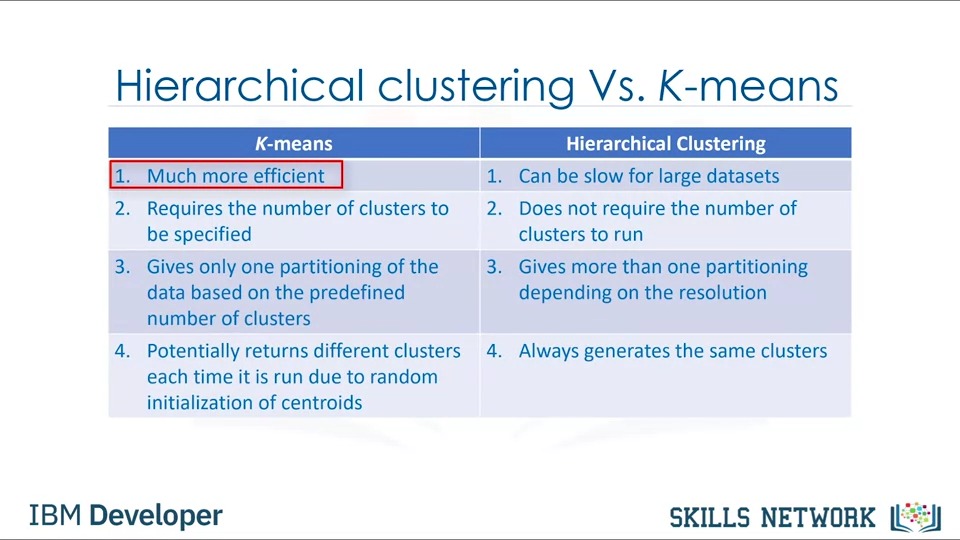


Now the question is, how can we calculate the distance between clusters when there are multiple patients in each cluster? We can use different criteria to find the closest clusters and merge them.



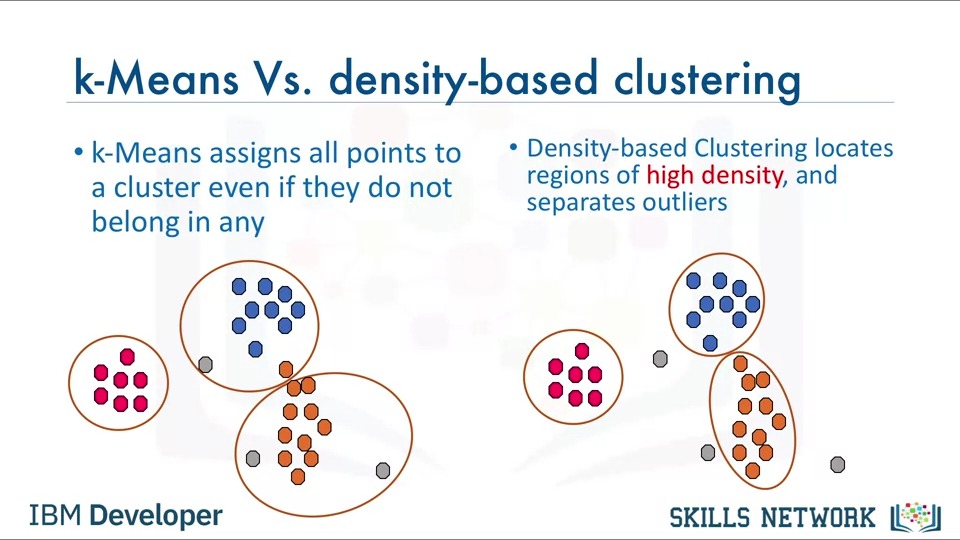




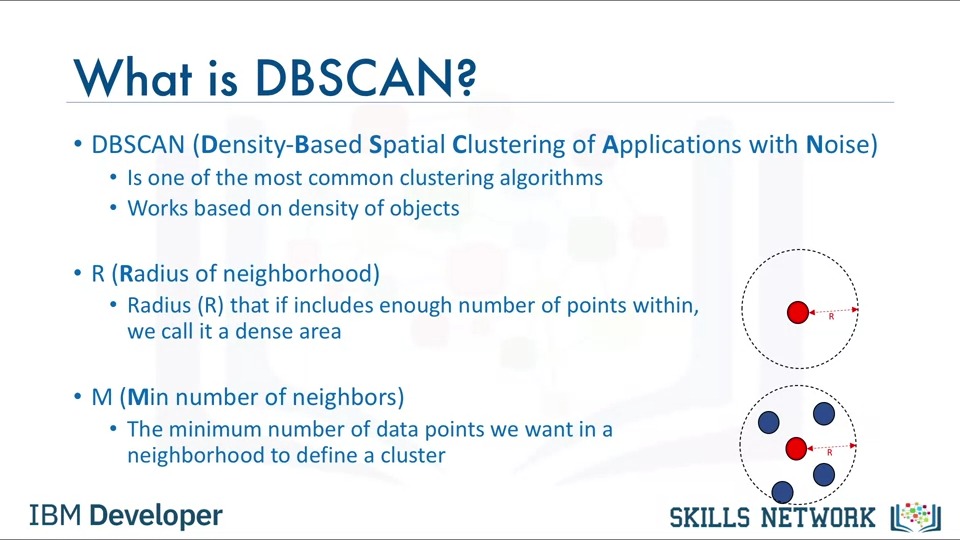


## **Clustering: Density-based Clustering (DBSCAN)**

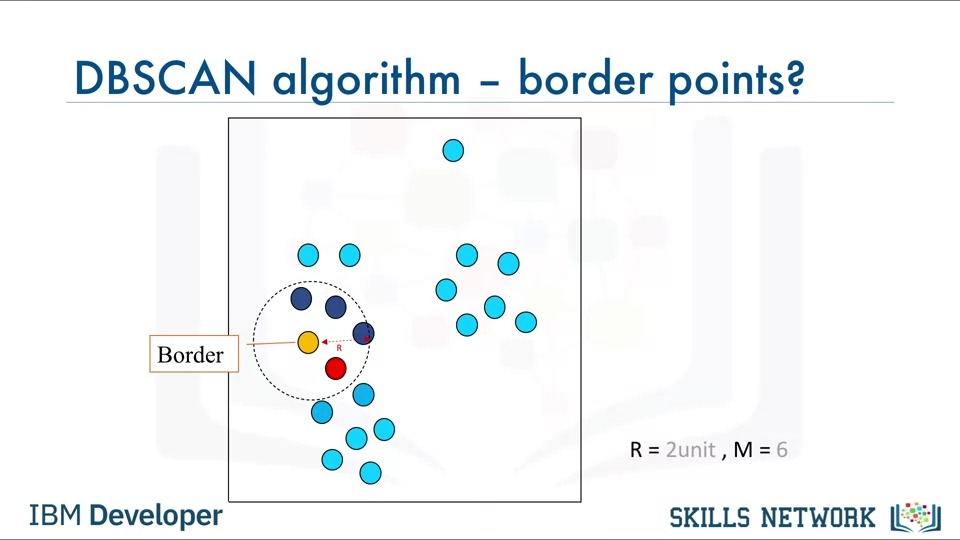


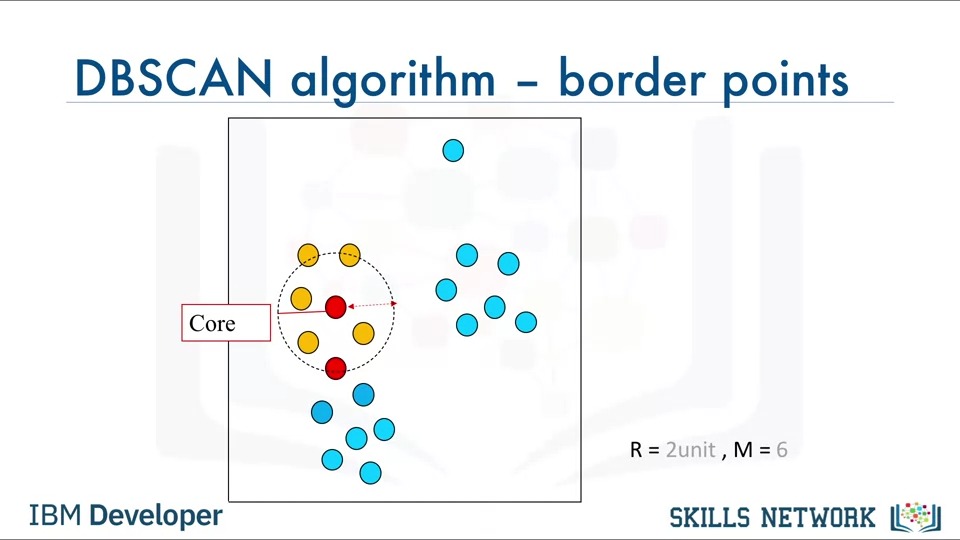


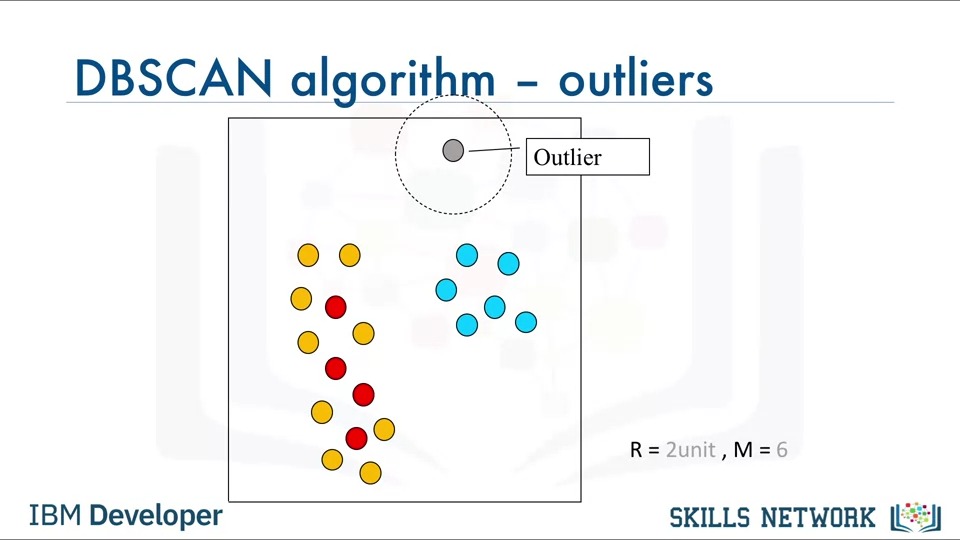
M determines the minimum number of data points we want in a neighborhood to define a cluster.



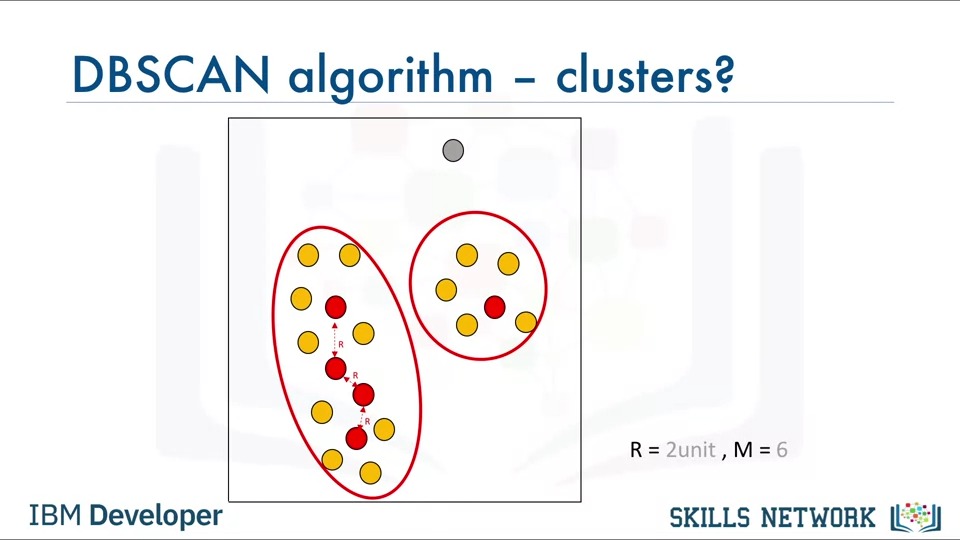
What is a border point? A data point is a border point if A; its neighbourhood contains less than M data points or B; it is reachable from some core point.

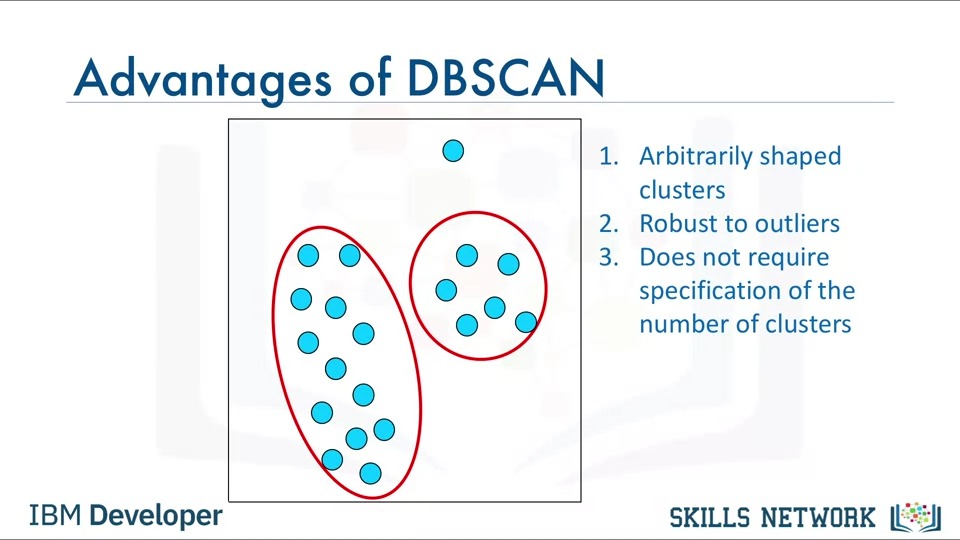


 What is an outlier? An outlier is a point that is not a core point and also is not close enough to be reachable from a core point.

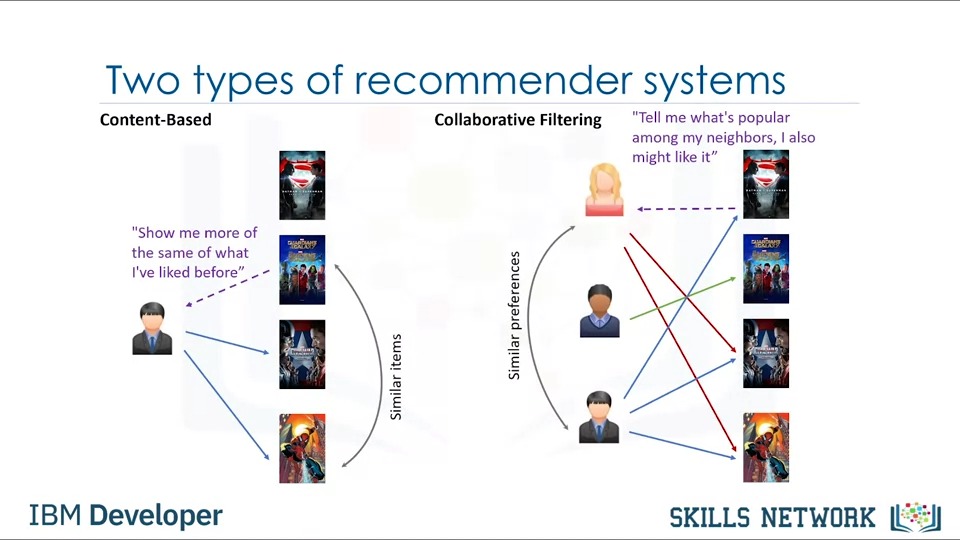


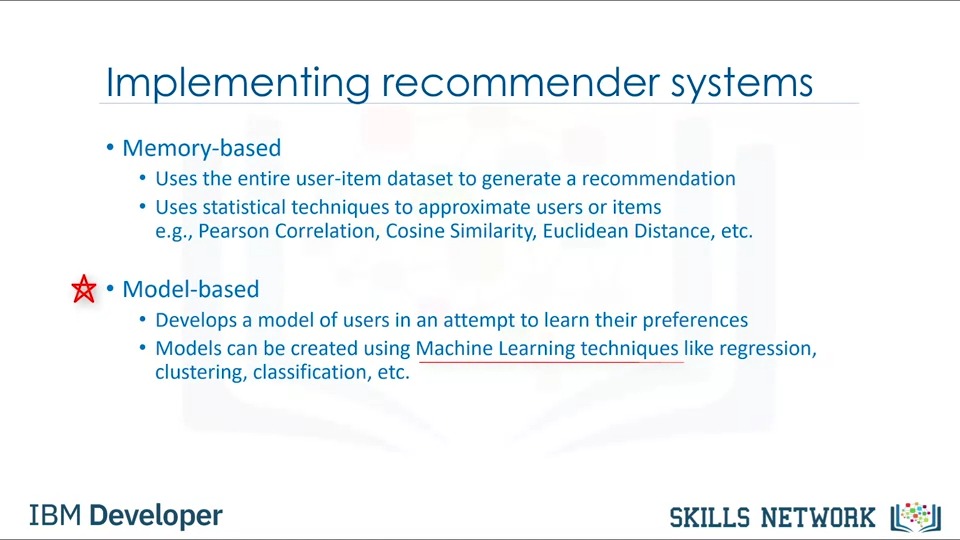
So, a cluster is formed as at least one core point plus all reachable core points plus all their borders.



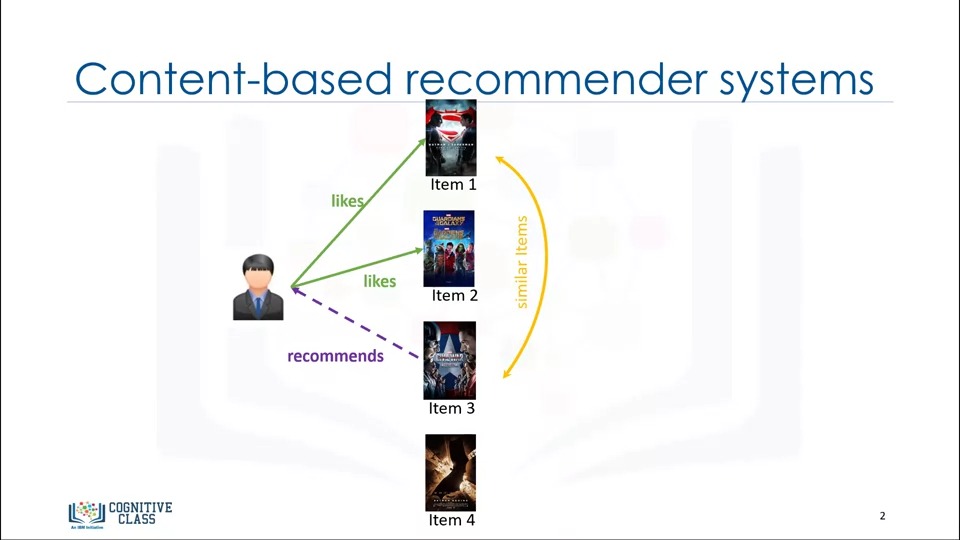


# **Recommendation**

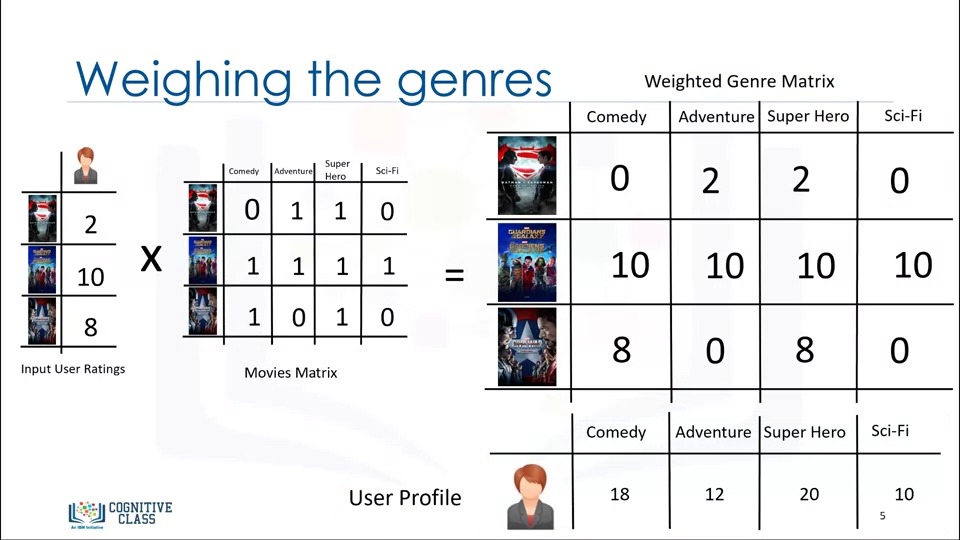


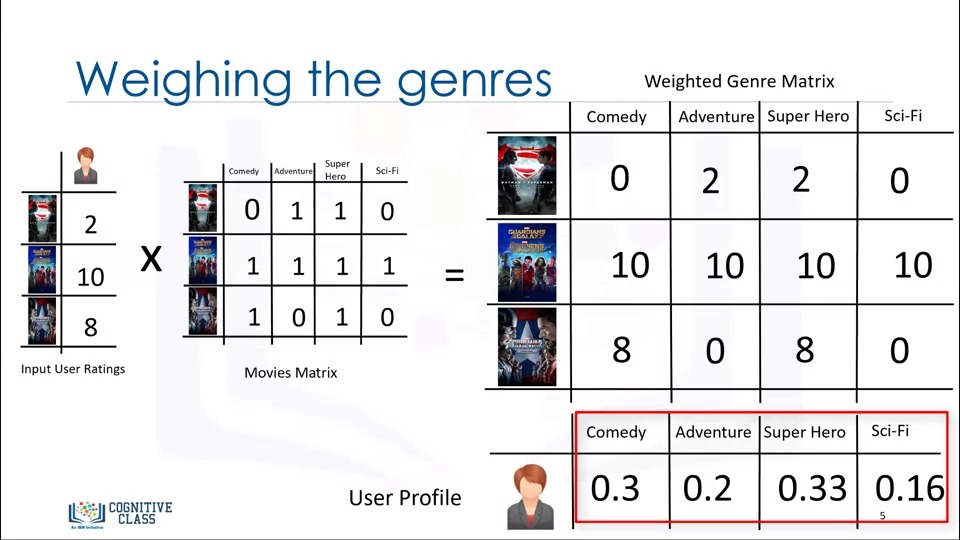


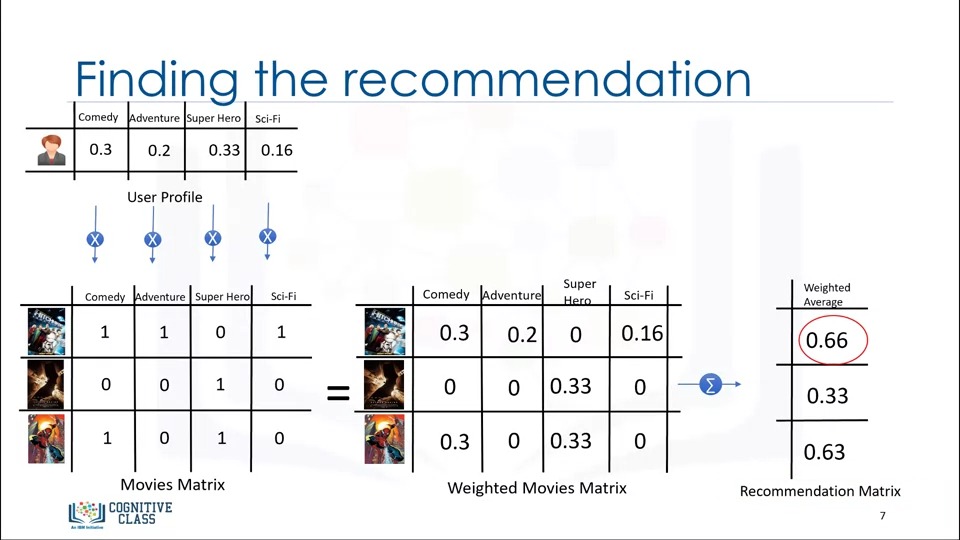
## **Content-based Recommendation**



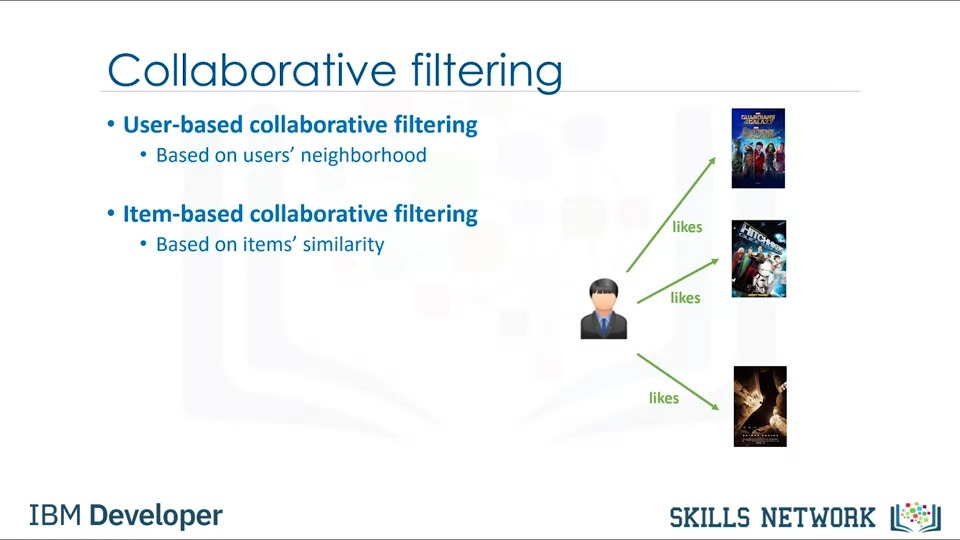


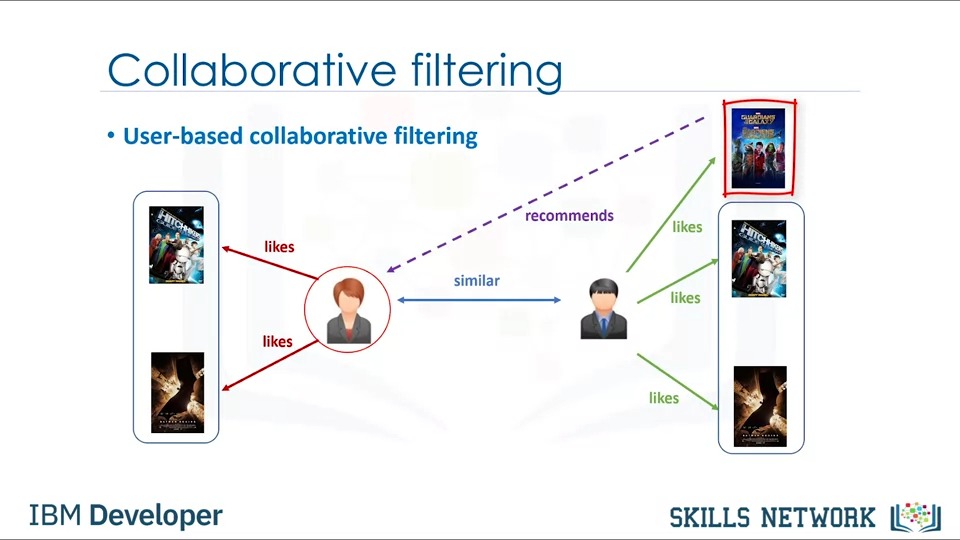




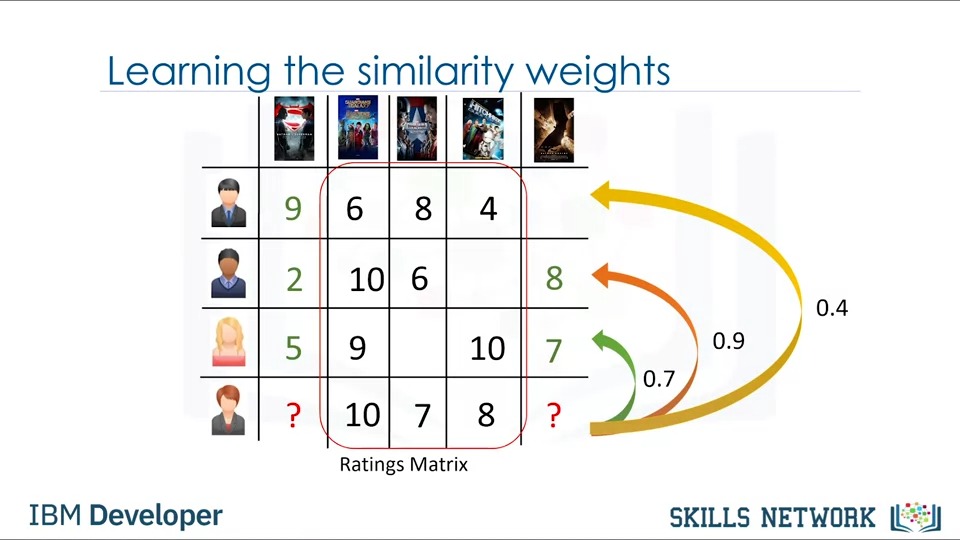


## **Collaborative Filtering Recommendation**



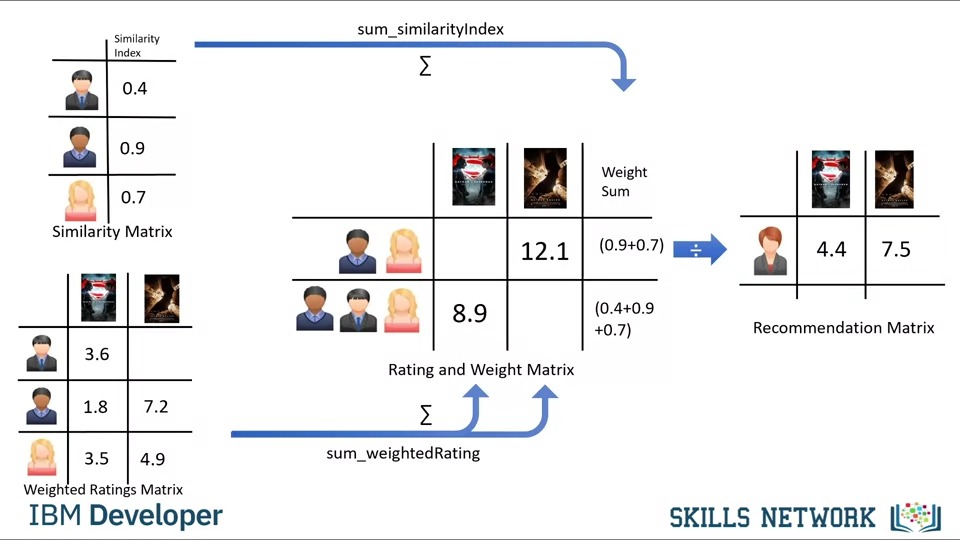


Regardless of what we use for similarity measurement, let's say for example, the similarity could be 0.7, 0.9, and 0.4 between the active user and other users.



In fact, it incorporates the behavior of other users and gives more weight to the ratings of those users who are more similar to the active user.





Therefore, the recommendations here are based on the items in the neighborhood that a user might prefer.

