**WEEK 9**

## Unsupervised Learning

* Unsupervised learning is a machine learning technique in which models are not supervised using training dataset.
* Instead, models itself find the hidden patterns and insights from the given data.
* Unsupervised learning is a type of machine learning in which models are trained using unlabelled dataset and are allowed to act on that data without any supervision.
* Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data.
* The goal of unsupervised learning is to **find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format.**

**Importance of Unsupervised Learning:**

* Unsupervised learning is helpful for finding useful insights from the data.
* Unsupervised learning is much similar as a human learns to think by their own experiences, which makes it closer to the real AI.
* Unsupervised learning works on unlabeled and uncategorized data which make unsupervised learning more important.
* In real-world, we do not always have input data with the corresponding output so to solve such cases, we need unsupervised learning.

**Common unsupervised learning approaches**

**Clustering**

* Clustering is a method of grouping the objects into clusters such that objects with most similarities remains into a group and has less or no similarities with the objects of another group.
* Cluster analysis finds the commonalities between the data objects and categorizes them as per the presence and absence of those commonalities.

**Association**

* An association rule is an unsupervised learning method which is used for finding the relationships between variables in the large database.
* It determines the set of items that occurs together in the dataset. Association rule makes marketing strategy more effective.
* Such as people who buy X item (suppose a bread) are also tend to purchase Y (Butter/Jam) item.
* A typical example of Association rule is Market Basket Analysis.

**Dimensionality reduction**

* While more data generally yields more accurate results, it can also impact the performance of machine learning algorithms (e.g., overfitting) and it can also make it difficult to visualize datasets.
* Dimensionality reduction is a technique used when the number of features, or dimensions, in a given dataset is too high.
* It reduces the number of data inputs to a manageable size while also preserving the integrity of the dataset as much as possible.

**Advantages of Unsupervised Learning**

* Unsupervised learning is used for more complex tasks as compared to supervised learning because of unlabeled input data.
* Unsupervised learning is preferable as it is easy to get unlabeled data in comparison to labeled data.

**Challenges associated with Unsupervised Learning**

* Unsupervised learning is intrinsically more difficult than supervised learning as it does not have corresponding output.
* The result of the unsupervised learning algorithm might be less accurate as input data is not labeled, and algorithms do not know the exact output in advance
* Number of clusters are normally not known beforehand.
* For partitional clustering algorithms, such as K-means, different initial centers may lead to different clustering results, moreover K is unknown.
* Time complexity - partitional clustering algorithms are O(N) whereas hierarchical are O(N2).
* The similarity criteria is not clear - should we use Euclidean or cosine or Tanimoto or Mahalanobis distance or whatever?
* In hierarchical clustering, at what stage we should stop is not clear.
* Evaluating clustering results are difficult because labels are not available at the beginning.

**Clustering Types**

The clustering methods are broadly divided into **Hard clustering** (datapoint belongs to only one group) and **Soft Clustering** (data points can belong to another group also). But there are also other various approaches of Clustering exist. Below are the main clustering methods used in Machine learning:

1. **Partitioning Clustering**
2. **Density-Based Clustering**
3. **Distribution Model-Based Clustering**
4. **Hierarchical Clustering**
5. **Fuzzy Clustering**

**K-Means Clustering Algorithm**

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science.

**What is K-Means Algorithm?**

* K-Means Clustering is an [Unsupervised Learning algorithm](https://www.javatpoint.com/unsupervised-machine-learning), which groups the unlabelled dataset into different clusters.
* Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

“It is an iterative algorithm that divides the unlabelled dataset into *k* different clusters in such a way that each dataset belongs only one group that has similar properties.”

* It is a centroid-based algorithm, where each cluster is associated with a centroid.
* The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

The algorithm takes the unlabelled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means [clustering](https://www.javatpoint.com/clustering-in-machine-learning) algorithm mainly performs two tasks:

* Determines the best value for K centre points or centroids by an iterative process.
* Assigns each data point to its closest k-centre. Those data points which are near to the particular k-centre, create a cluster.



**Working of K-means**

The working of the K-Means algorithm is explained in the below steps:

**Step-1:** Select the number K to decide the number of clusters.

**Step-2:** Select random K points or centroids. (It can be other from the input dataset).

**Step-3:** Assign each data point to their closest centroid, which will form the predefined K clusters.

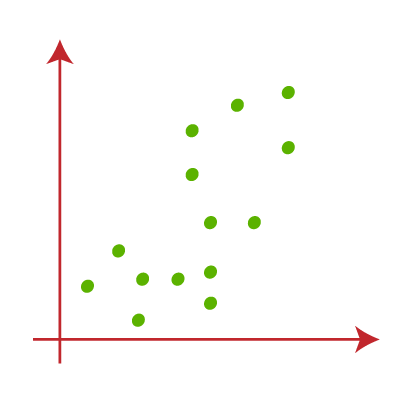
**Step-4:** Calculate the variance and place a new centroid of each cluster.

**Step-5:** Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

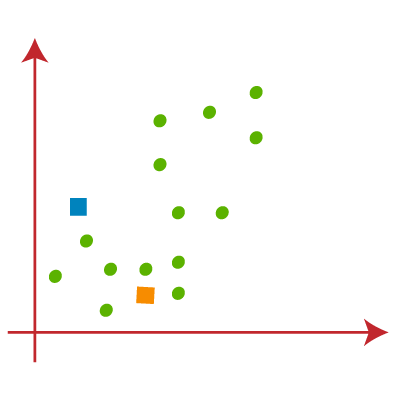
**Step-6:** If any reassignment occurs, then go to step-4 else go to FINISH.

**Step-7**: The model is ready.

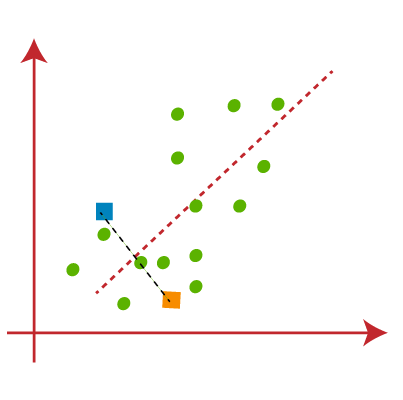
Suppose we have two variables M1 and M2. The x-y axis scatter plot of these two variables is given below:



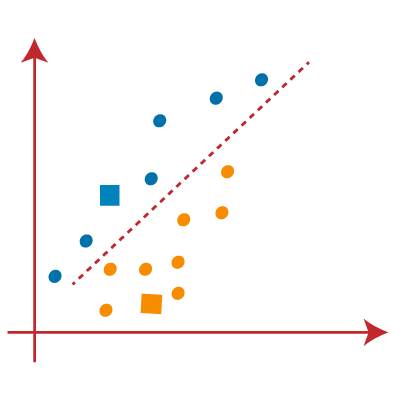
* Let's take number k of clusters, i.e., K=2, to identify the dataset and to put them into different clusters. It means here we will try to group these datasets into two different clusters.
* We need to choose some random k points or centroid to form the cluster. These points can be either the points from the dataset or any other point. So, here we are selecting the below two points as k points, which are not the part of our dataset. Consider the below image:



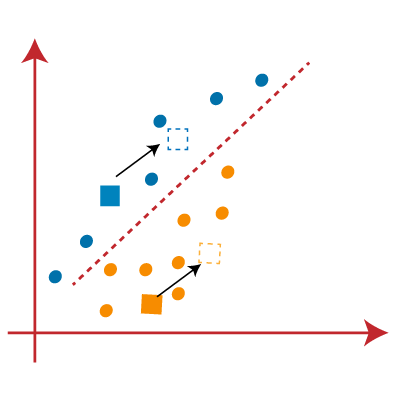
* Now we will assign each data point of the scatter plot to its closest K-point or centroid. We will compute it by applying some mathematics that we have studied to calculate the distance between two points. So, we will draw a median between both the centroids. Consider the below image:



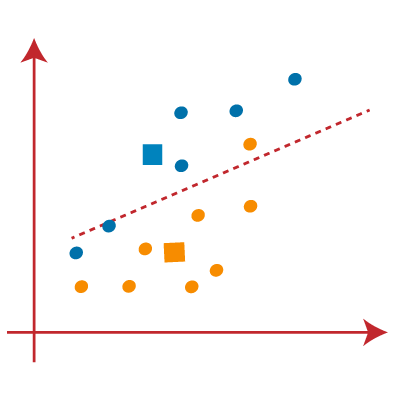
* From the above image, it is clear that points left side of the line is near to the K1 or blue centroid, and points to the right of the line are close to the yellow centroid. Let's color them as blue and yellow for clear visualization.



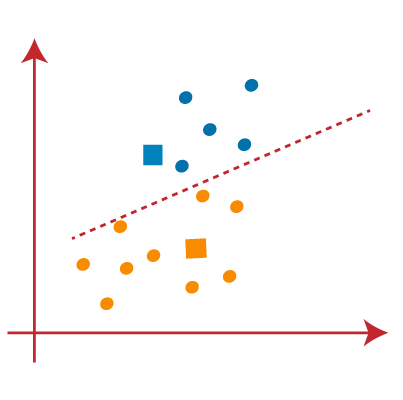
* As we need to find the closest cluster, so we will repeat the process by choosing **a new centroid**. To choose the new centroids, we will compute the center of gravity of these centroids, and will find new centroids as below:



* Next, we will reassign each datapoint to the new centroid. For this, we will repeat the same process of finding a median line. The median will be like below image:

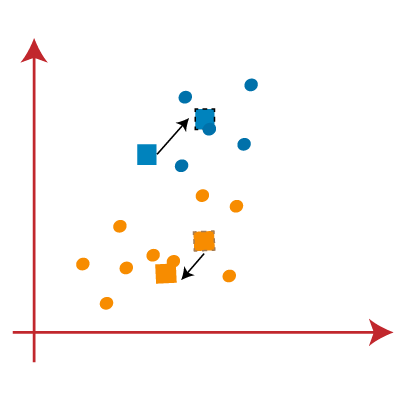


* From the above image, we can see, one yellow point is on the left side of the line, and two blue points are right to the line. So, these three points will be assigned to new centroids.

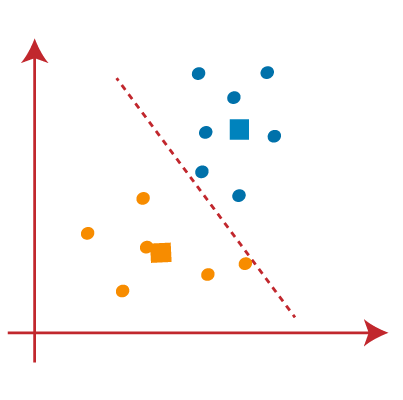


As reassignment has taken place, so we will again go to the step-4, which is finding new centroids or K-points.

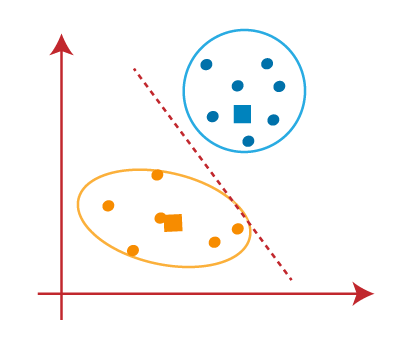
* We will repeat the process by finding the centre of gravity of centroids, so the new centroids will be as shown in the below image:



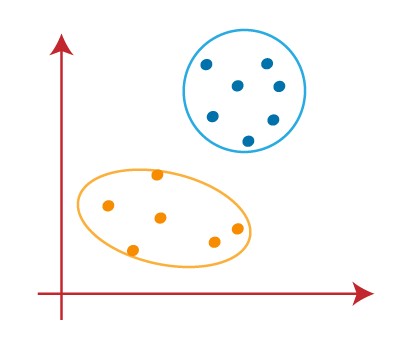
* As we got the new centroids so again will draw the median line and reassign the data points. So, the image will be:



* We can see in the above image; there are no dissimilar data points on either side of the line, which means our model is formed. Consider the below image:



* As our model is ready, so we can now remove the assumed centroids, and the two final clusters will be as shown in the below image:

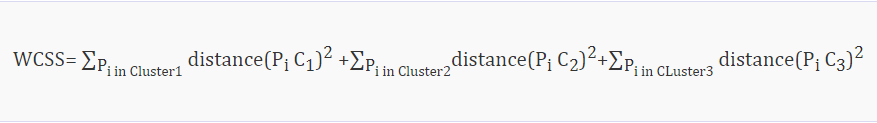


**How to Choose the Right Number of Clusters?**

The performance of the K-means clustering algorithm depends upon highly efficient clusters that it forms. But choosing the optimal number of clusters is a big task. There are some different ways to find the optimal number of clusters, but here we are discussing the most appropriate method to find the number of clusters or value of K. The method is given below:

**Elbow Method**

The Elbow method is one of the most popular ways to find the optimal number of clusters. This method uses the concept of WCSS value. **WCSS** stands for **Within Cluster Sum of Squares**, which defines the total variations within a cluster. The formula to calculate the value of WCSS (for 3 clusters) is given below:



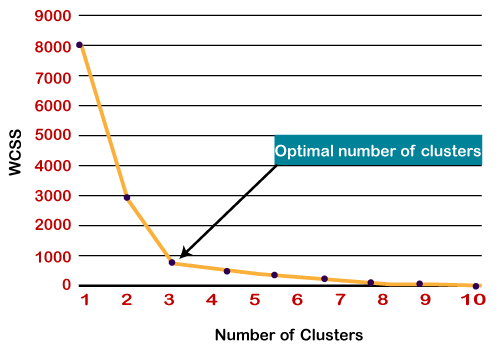
**In the above formula of WCSS,**

* ∑Pi in Cluster1 distance(Pi C1)2: It is the sum of the square of the distances between each data point and its centroid within a cluster1 and the same for the other two terms.
* To measure the distance between data points and centroid, we can use any method such as Euclidean distance or Manhattan distance.

To find the optimal value of clusters, the elbow method follows the below steps:

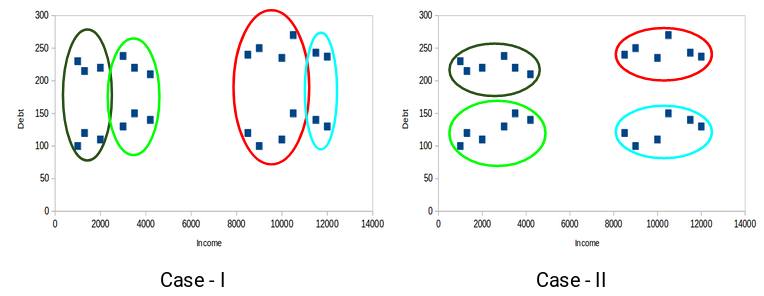
* It executes the K-means clustering on a given dataset for different K values (ranges from 1-10).
* For each value of K, calculates the WCSS value.
* Plots a curve between calculated WCSS values and the number of clusters K.
* The sharp point of bend or a point of the plot looks like an arm, then that point is considered as the best value of K.

Since the graph shows the sharp bend, which looks like an elbow, hence it is known as the elbow method. The graph for the elbow method looks like the below image:



## Evaluation Metrics for Clustering

The primary aim of clustering is not just to make clusters, but to make good and meaningful ones.

[](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/08/Screenshot-from-2019-08-08-14-51-31.png)

Here, we used only two features and hence it was easy for us to visualize and decide which of these clusters is better.

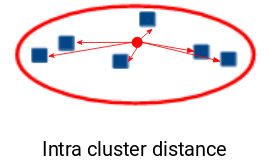
Unfortunately, that’s not how real-world scenarios work. We will have a ton of features to work with. Let’s take the customer segmentation example – we will have features like customer’s income, occupation, gender, age, and many more. Visualizing all these features together and deciding better and meaningful clusters would not be possible for us.

This is where we can make use of evaluation metrics. Let’s discuss a few of them and understand how we can use them to evaluate the quality of our clusters.

### Inertia

Inertia tells us how far the points within a cluster are. So,**inertia actually calculates the sum of distances of all the points within a cluster from the centroid of that cluster.**

We calculate this for all the clusters and the final inertia value is the sum of all these distances. This distance within the clusters is known as **intra-cluster distance**. So, inertia gives us the sum of intra-cluster distances:

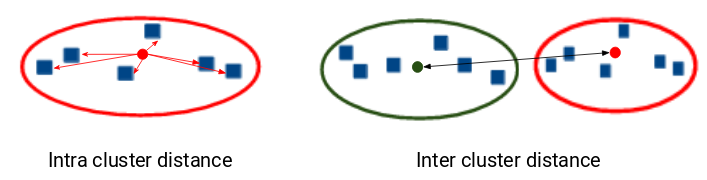
[](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/08/Screenshot-from-2019-08-08-15-32-17.png)

We want the points within the same cluster to be similar to each other. Hence, **the distance between them should be as low as possible**. We can say that the lesser the inertia value, the better our clusters are.

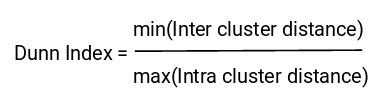
### Dunn Index

We now know that inertia tries to minimize the intra-cluster distance. It is trying to make more compact clusters.

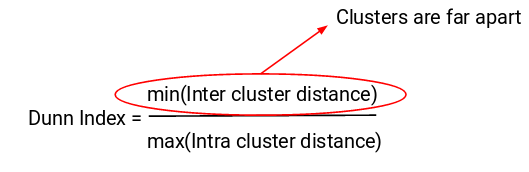
If the distance between the centroid of a cluster and the points in that cluster is small, it means that the points are closer to each other. So, inertia makes sure that the first property of clusters is satisfied. But it does not care about the second property – that different clusters should be as different from each other as possible. This is where Dunn index can come into action.

[](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/08/Screenshot-from-2019-08-08-15-37-02.png)

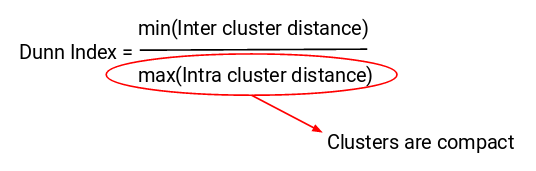
Along with the distance between the centroid and points,**the Dunn index also takes into account the distance between two clusters**. This distance between the centroids of two different clusters is known as**inter-cluster distance**. Let’s look at the formula of the Dunn index:

[](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/08/Screenshot-from-2019-08-08-15-37-22.png)

* Dunn index is the ratio of the minimum of inter-cluster distances and maximum of intra-cluster distances.
* We want to maximize the Dunn index. The more the value of the Dunn index, the better will be the clusters.

[](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/08/Screenshot-from-2019-08-08-15-39-19.png)

In order to maximize the value of the Dunn index, the numerator should be maximum. Here, we are taking the minimum of the inter-cluster distances. So, the distance between even the closest clusters should be more which will eventually make sure that the clusters are far away from each other.

[](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/08/Screenshot-from-2019-08-08-15-39-34.png)

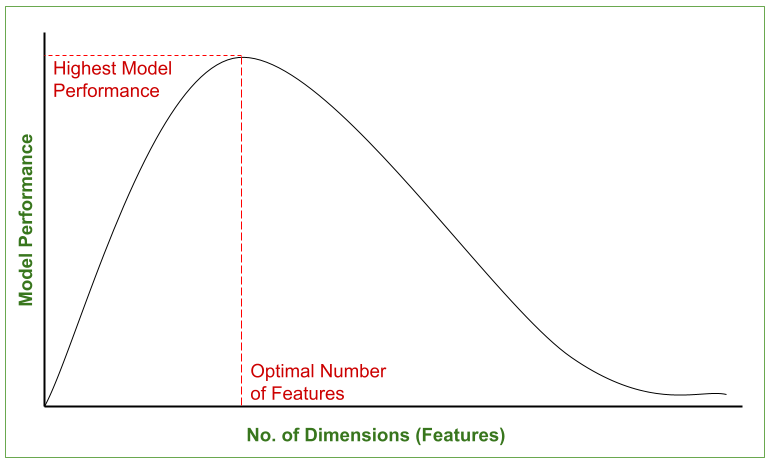
Also, the denominator should be minimum to maximize the Dunn index. Here, we are taking the maximum of intra-cluster distances. Again, the intuition is the same here. The maximum distance between the cluster centroids and the points should be minimum which will eventually make sure that the clusters are compact.

**Dimensionality Reduction**

In machine learning, “Dimensionality” simply refers to the number of features/variables in your dataset. Dimensionality Reduction is a statistical/ML-based technique wherein we try to reduce the number of features in our dataset and obtain a dataset with an optimal number of dimensions.

When the number of features/variables is very large relative to the number of observations in your dataset, some algorithms struggle to train effective models. This is called the **“Curse of Dimensionality.”**

The following graph represents the change in model performance with the increase in the number of dimensions of the dataset. It can be observed that the model performance is best only at an option dimension, beyond which it starts decreasing.



**Why do we need Dimensionality Reduction?**

* High dimensional data often leads to over-fitting when learning a model, which means that the model will perform well on the training data but poorly on test data.
* The higher the number of features, the harder it gets to visualize the training set and then work on it.
* Sometimes, most of these features are correlated, and hence redundant. This is where dimensionality reduction algorithms come into play.
* It reduces the time and storage space required.
* It helps Remove multi-collinearity which improves the interpretation of the parameters of the machine learning model.
* It becomes easier to visualize the data when reduced to very low dimensions such as 2D or 3D.
* It avoids the curse of dimensionality.
* It removes irrelevant features from the data, Because having irrelevant features in the data can decrease the accuracy of the models and make your model learn based on irrelevant features.

**Types of Dimensionality reduction:**

**Feature Selection-**

Feature selection is a method introduced in the machine learning in order to remove less significant features from the data, so that the model is trained only on the features that contribute most to the prediction(dependent) variable.

Types of Feature Selection methods are:

\* Filter

\* Wrapper

\* Embedded

**Feature Extraction-**

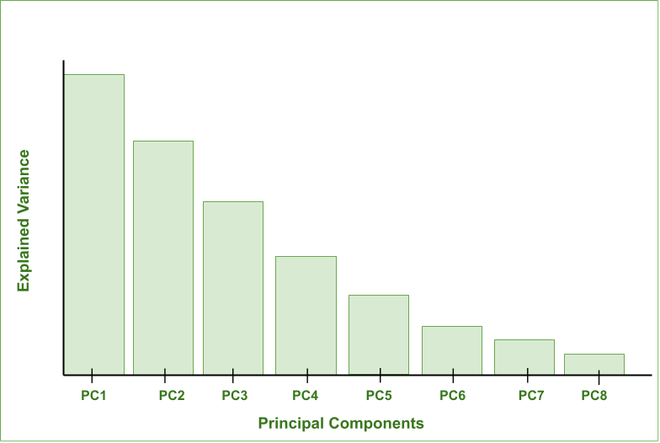
Feature extraction is a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing.

This process is generally done with image and text data, where only the important feature is extracted and sent ahead for processing instead of taking the whole data.

Feature Extraction can be Unsupervised (PCA) or supervised (LDA).

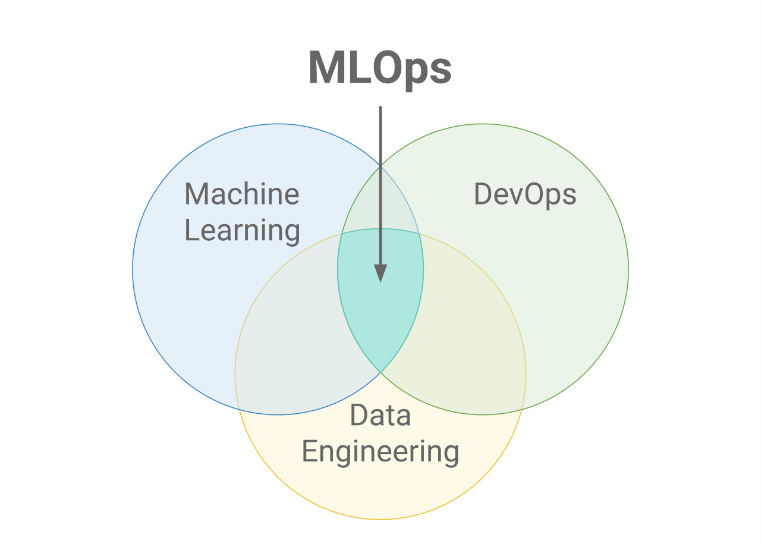
**Dimensionality Reduction using PCA**

* Principal Component Analysis is a technique of feature extraction that maps a higher dimensional feature space to a lower-dimensional feature space.
* While reducing the number of dimensions, PCA ensures that maximum information of the original dataset is retained in the dataset with the reduced no. of dimensions and the co-relation between the newly obtained Principal Components is minimum.
* The new features obtained after applying PCA are called **Principal Components** and are denoted as PCi (i=1,2,3…n).
* Here, (Principal Component-1) PC1 captures the maximum information of the original dataset, followed by PC2, then PC3 and so on.
* The following bar graph depicts the amount of Explained Variance captured by various Principal Components.



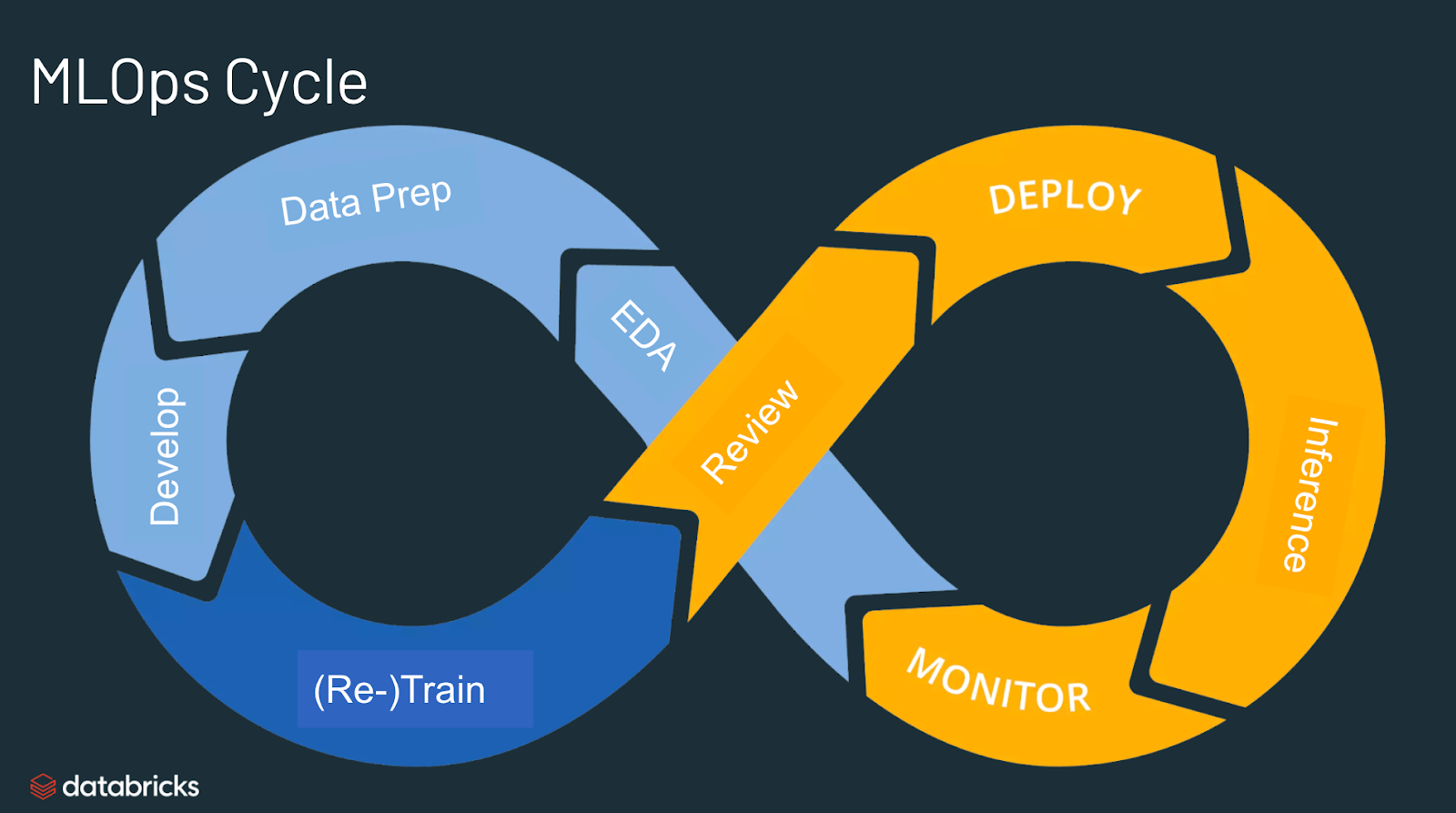
**MLOps**

* MLOps stands for Machine Learning Operations.
* MLOps is a core function of Machine Learning engineering, focused on streamlining the process of taking machine learning models to production, and then maintaining and monitoring them.
* MLOps is a collaborative function, often comprising data scientists, devops engineers, and IT.
* MLOps can encompass everything from the data pipeline to machine learning model production.



**Why MLOps?**

* Productionizing machine learning is difficult.
* The machine learning lifecycle consists of many complex components such as data ingest, data prep, model training, model tuning, model deployment, model monitoring, and much more.
* It also requires collaboration and hand-offs across teams, from Data Engineering to Data Science to ML Engineering.
* Naturally, it requires stringent operational rigor to keep all these processes synchronous and working in tandem. MLOps encompasses the experimentation, iteration, and continuous improvement of the machine learning lifecycle.



**The primary benefits of MLOps**

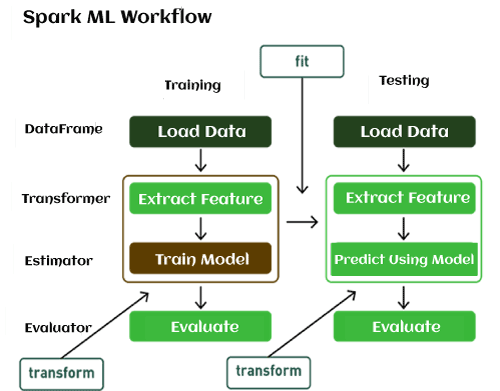
**Efficiency:** MLOps allows data teams to achieve faster model development, deliver higher quality ML models, and faster deployment and production.

**Scalability:** MLOps also enables vast scalability and management where thousands of models can be overseen, controlled, managed, and monitored for continuous integration, continuous delivery, and continuous deployment. Specifically, MLOps provides reproducibility of ML pipelines, enabling more tightly-coupled collaboration across data teams, reducing conflict with devops and IT, and accelerating release velocity.

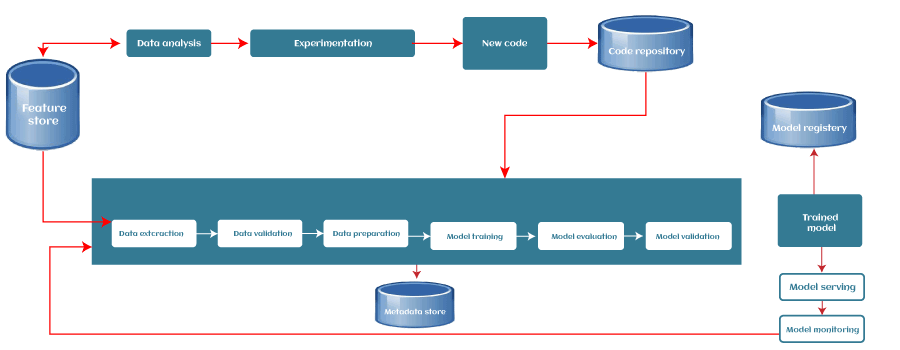
**Risk reduction:** Machine learning models often need regulatory scrutiny and drift-check, and MLOps enables greater transparency and faster response to such requests and ensures greater compliance with an organization’s or industry’s policies.

**ML pipeline:**

* **A Machine Learning pipeline is a process of automating the workflow of a complete machine learning task**.
* It can be done by enabling a sequence of data to be transformed and correlated together in a model that can be analyzed to get the output.
* A typical pipeline includes raw data input, features, outputs, model parameters, ML models, and Predictions.
* ML Pipeline contains multiple sequential steps that perform everything ranging from data extraction and pre-processing to model training and deployment in Machine learning in a modular approach.
* **In the pipeline, each step is designed as an independent module, and all these modules are tied together to get the final result.**
* A typical pipeline contains various stages. However, there are two main pipeline stages:



1. **Transformer:** It takes a dataset as an input and creates an augmented dataset as output. For example, A tokenizer works as Transformer, which takes a text dataset, and transforms it into tokenized words.
2. **Estimator:** An estimator is an algorithm that fits on the input dataset to generate a model, which is a transformer. For example, regression is an Estimator that trains on a dataset with labels and features and produces a logistic regression model.



**1. Data Ingestion**

Each ML pipeline starts with the Data ingestion step. In this step, the data is processed into a well-organized format, which could be suitable to apply for further steps.

**2. Data Validation**

Data validation focuses on statistics of the new data, e.g., range, number of categories, distribution of categories, etc. In this step, data scientists can detect if any anomaly present in the data.

**3. Data Pre-processing**

The pre-processing step involves preparing the raw data and making it suitable for the ML model. The process includes Data cleaning, feature scaling, etc. The product or output of the data pre-processing step becomes the final dataset that can be used for model training and testing.

**4. Model Training & Tuning**

In this step, the model is trained to take the input (pre-processed dataset) and predicts an output with the highest possible accuracy.

However, there could be some difficulties with larger models or with large training data sets. So, for this, efficient distribution of the model training or model tuning is required.

This issue of the model training stage can be solved with pipelines as they are scalable, and a large number of models can be processed concurrently.

**5. Model Analysis**

After model training, need to determine the optimal set of parameters by using the loss of accuracy metrics. an in-depth analysis of the model's performance is crucial for the final version of the model.

**6. Model Versioning**

The model versioning step keeps track of which model, set of hyperparameters, and datasets have been selected as the next version to be deployed.

**7. Model Deployment**

After training and analyzing the model, it's time to deploy the model. An ML model can be deployed in three ways, which are:

* Using the Model server,
* In a Browser
* On Edge device

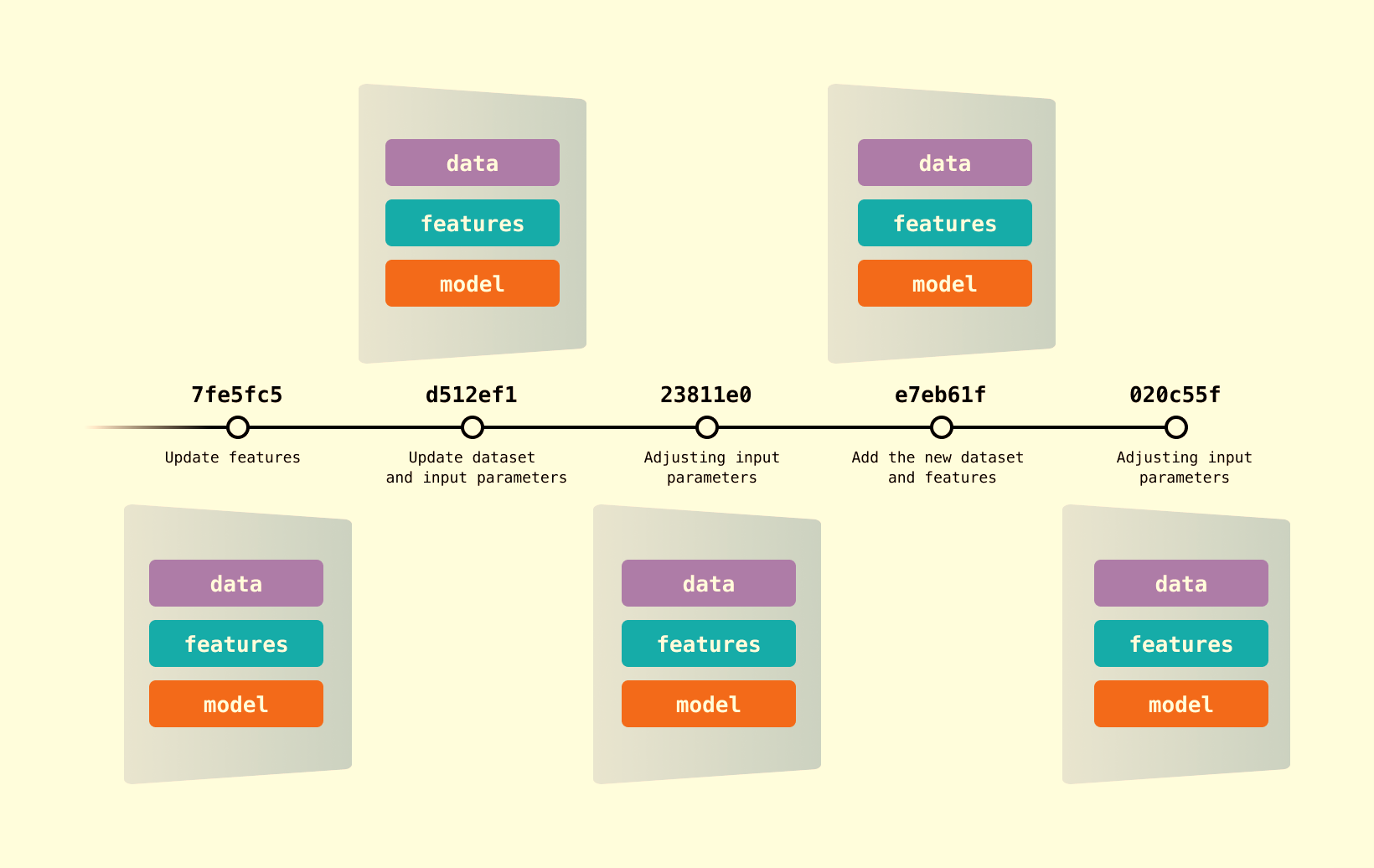
**8. Feedback Loop**

Each pipeline forms a closed-loop to provide feedback. With this close loop, data scientists can determine the effectiveness and performance of the deployed models. This step could be automated or manual depending on the requirement. **Except for the two manual review steps (the model analysis and the feedback step), we can automate the entire pipeline.**

**Versioning**

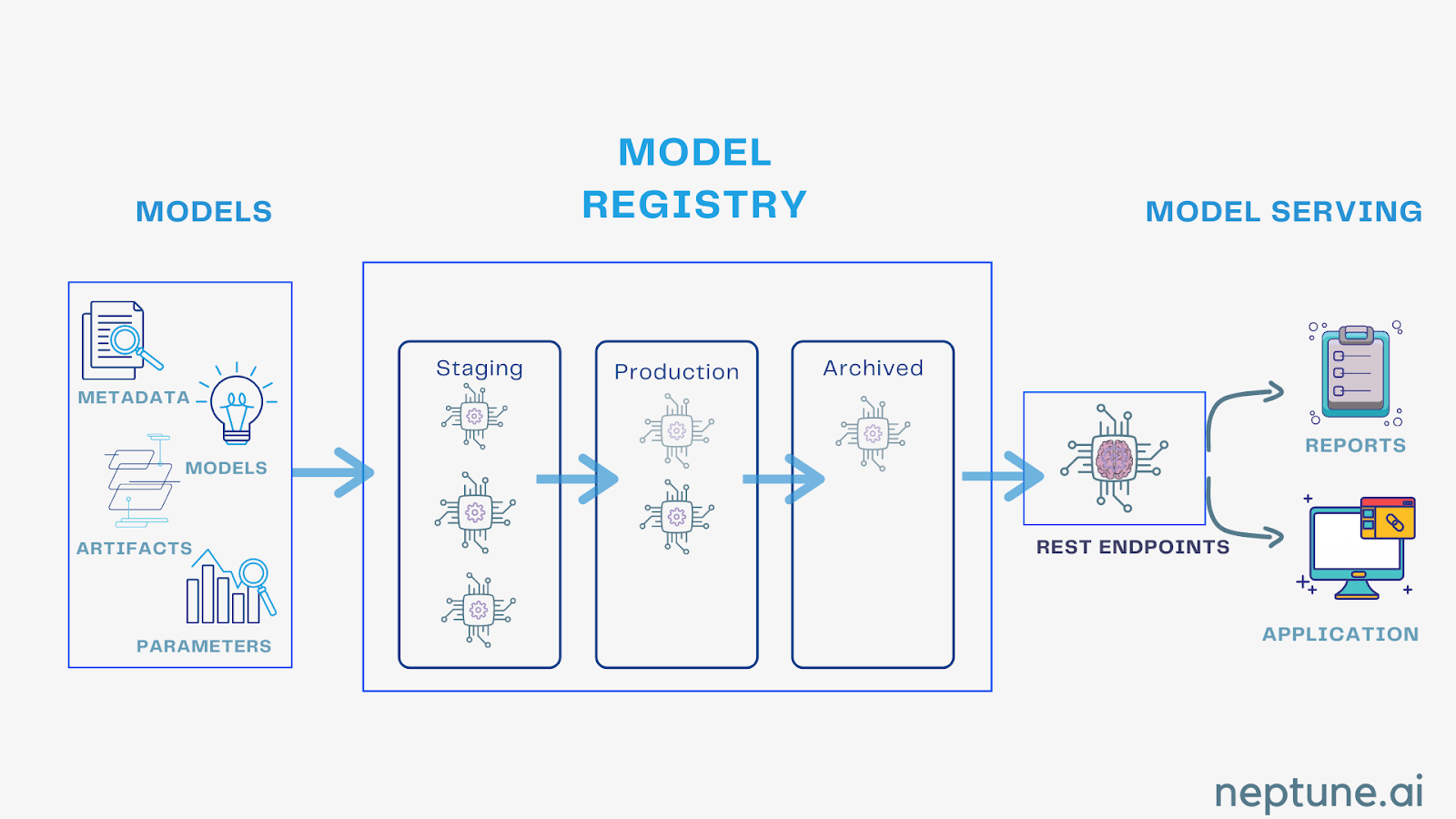
The use of code versioning tools is vital in the software development industry. The possibility of replicating the same code base so that several people can work on the same project simultaneously is a great benefit. In addition, versioning these bases allows them to work in different sections in an organized manner and without compromising the integrity of the code in production.

In a machine learning project, data scientists are continuously working on the development of new models. This process relies on trying different combinations of data, parameters, and algorithms. It's extremely positive to create an environment where it's possible to go back and forth on older or new experiments.



**Model registry**

The Model Registry is a system that allows machine learning engineers and data scientists to publish, test, monitor, govern and share them for collaboration with other teams. Essentially, the model registry is used when we done with our experimentation phase, and ready to share with the team and stakeholders.



**Monitoring**

ML model monitoring is the practice of tracking the performance of ML models in production to identify potential issues that can add negative business value. These practices help proactively monitor prediction quality issues, data relevance, model accuracy, and bias. ML monitoring constitutes the subset of AI observability where it showcases a bigger picture with testing, validation, and exploring unforeseen failure modes.

The performance of ML models starts degrading over time. It can be due to data inconsistencies, skews, and drifts, making deployed models inaccurate and irrelevant. Appropriate ML monitoring helps identify precisely when the model performance started diminishing. Such proactive monitoring helps take required actions like retraining models or replacing models. It helps foster users’ trust in ML systems.

**Deployment**

After training and analysing the model, it's time to deploy the model. An ML model can be deployed in three ways, which are:

* Using the Model server,
* In a Browser
* On Edge device