Model Tuning

Session Plan

1. Introduction
2. Discussion Questions on the concepts
3. Hands-on Case study
4. Extended Discussions and QnA
5. Summary

Discussion Questions

1. What is hyperparameter tuning?
2. Different types of hyperparameter tuning
3. How to use pipelines and Column Transformer to standardize the model building process

What is hyperparameter tuning?

- Hyperparameters are the parameters that govern the entire training process
- Their values are set before the learning process begins
- They have a significant effect on the model’s performance
- The process of finding optimal hyperparameters for a model is known as hyperparameter tuning
- Choosing optimal hyperparameters can lead to improvements in the overall model’s performance and can help in reducing both overfitting and underfitting

Different types of hyperparameter tuning

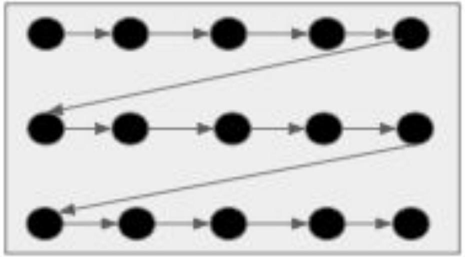
- Some models consist of a huge number of hyperparameters, and finding the optimal set of hyperparameters can be a very time-consuming process
- To make the process efficient, we'll look at 2 of the most common methods available in sklearn:
    - GridSearchCV
    - RandomizedSearchCV
- Grid search is best used when we have small search space, while Random search is best used when we have large search space
- We can use grid search to get the best possible results when we don't have any time constraints, but when we have time constraints, it's better to go with the random search
- Randomized search is known to give better results as compared to grid search

Grid Search

Grid search is a technique used to find the optimal set of hyperparameters for a model from the provided search space

Let’s understand how grid search works, with an example

- Let this grey box be set of all possible hyperparameters
- Let these black circles indicate the search space
- Grid search will iterate over all black circles in a sequence
- And finally gives the best set of hyperparameters based on the best score obtained

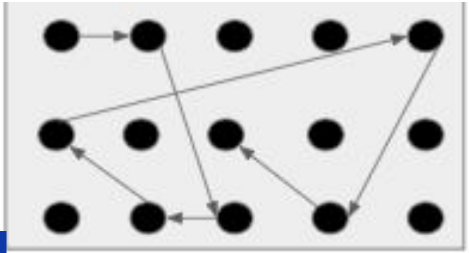


Doesn’t work well on large search spaces

It will find the best set of hyperparameters but with high cost

Random Search

- Random Search is another technique to find the best set of hyperparameters which takes lesser time than grid search
- Random search is very similar to grid search, the difference is that in the random search
  - we define **‘n_iter’ -** not all parameter values are tried out, but rather a fixed number of parameter settings is sampled from the specified distributions. The number of parameter settings that are tried is given by n_iter.
  - the set of hyperparameters is not searched sequentially
  - We can pass a range here instead of just numbers
- Out of the entire search space of hyperparameter, only n_iter number of set of hyperparameters will be checked **randomly**



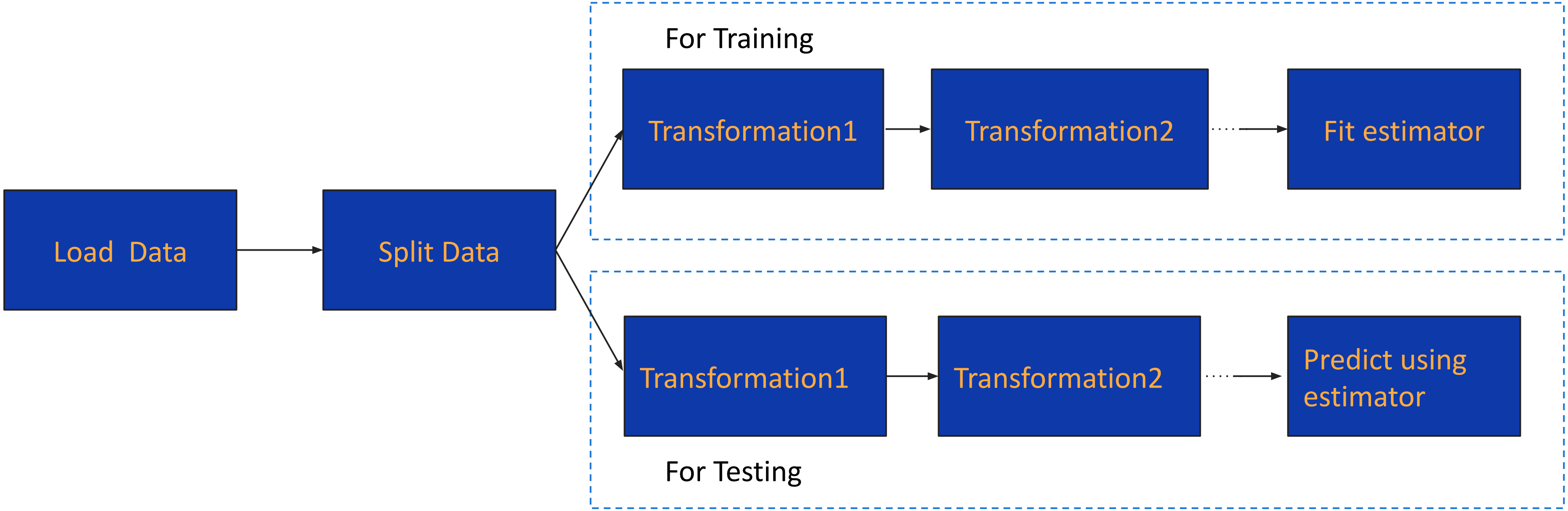
Works well on large search spaces
Gives better results than grid search

It doesn’t guarantee finding the best set of hyperparameters

Pipelines

- Almost always, we need to tie together many different data pre-processing steps, to prepare data for machine learning modeling
- It is paramount that the stages of transformation of data, represented by these processes, are standardized
- We can use use pipelines to sequentially apply a list of transformations for data preparation and fit an estimator at the end.

Without Pipelines



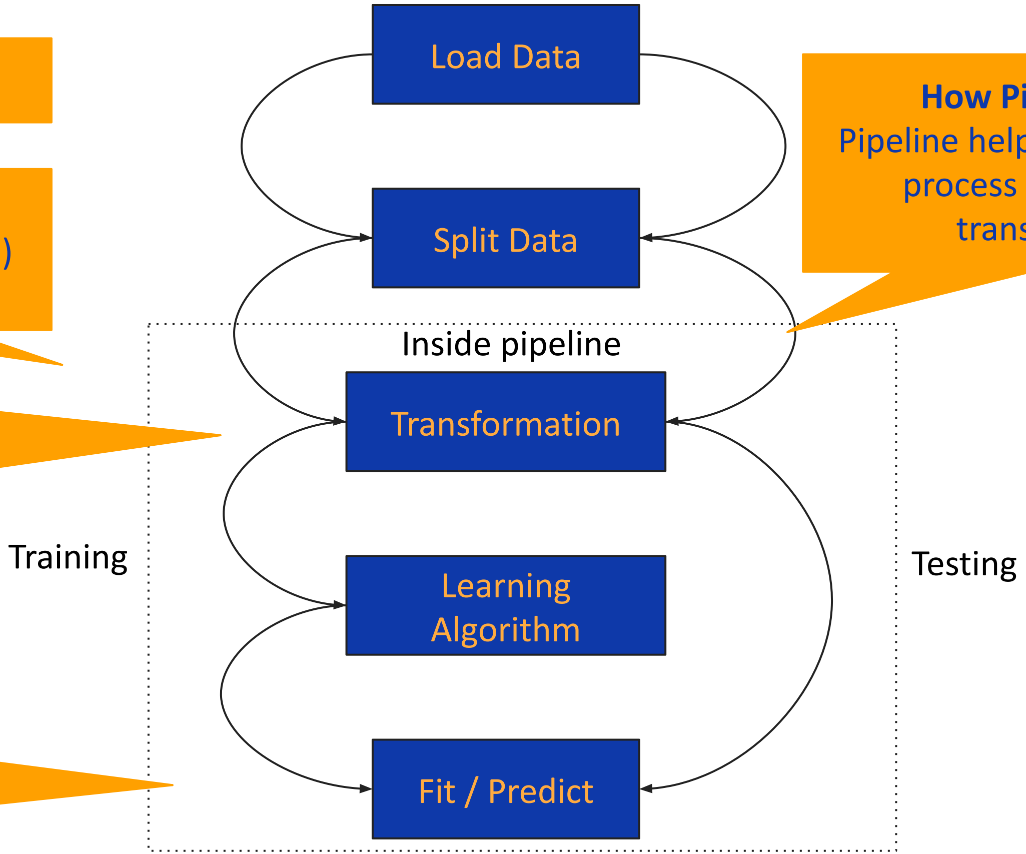
Using Pipelines

How Pipelines work?

All the steps inside a pipeline should have fit() and transform() except the last step

The transform step, transforms the data and transformed output of one step is input data for the next step

The last step of the pipeline can have either
fit() and predict() OR
fit() and transform()

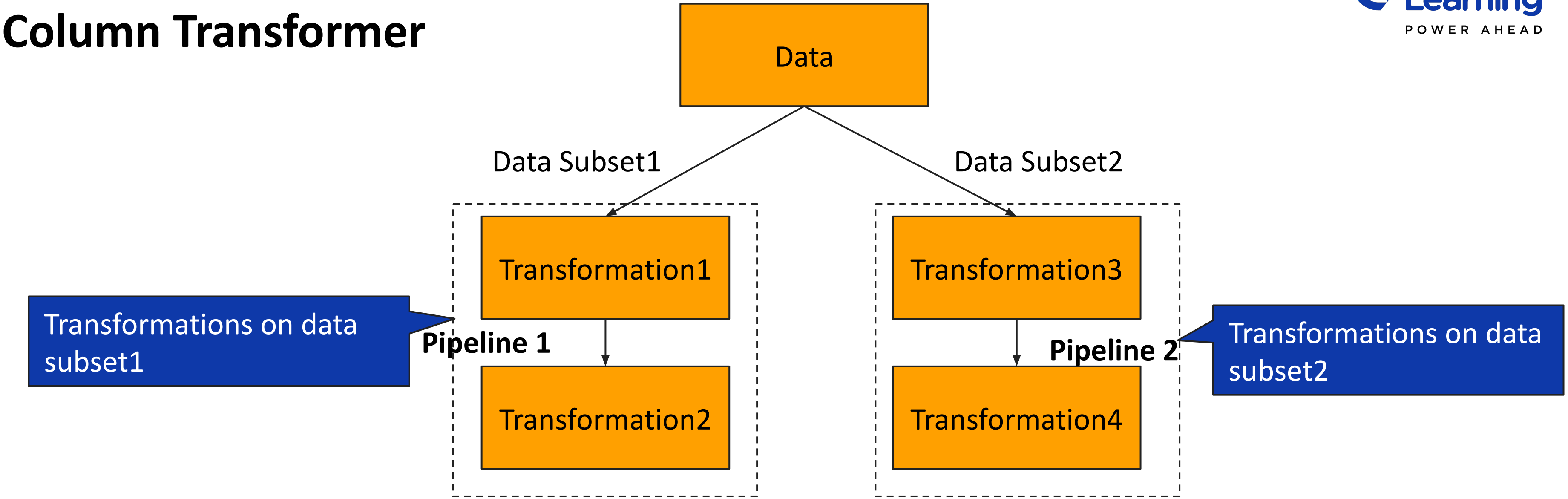


How Pipelines help?
Pipeline helps in simplifying the process of chaining the transformations

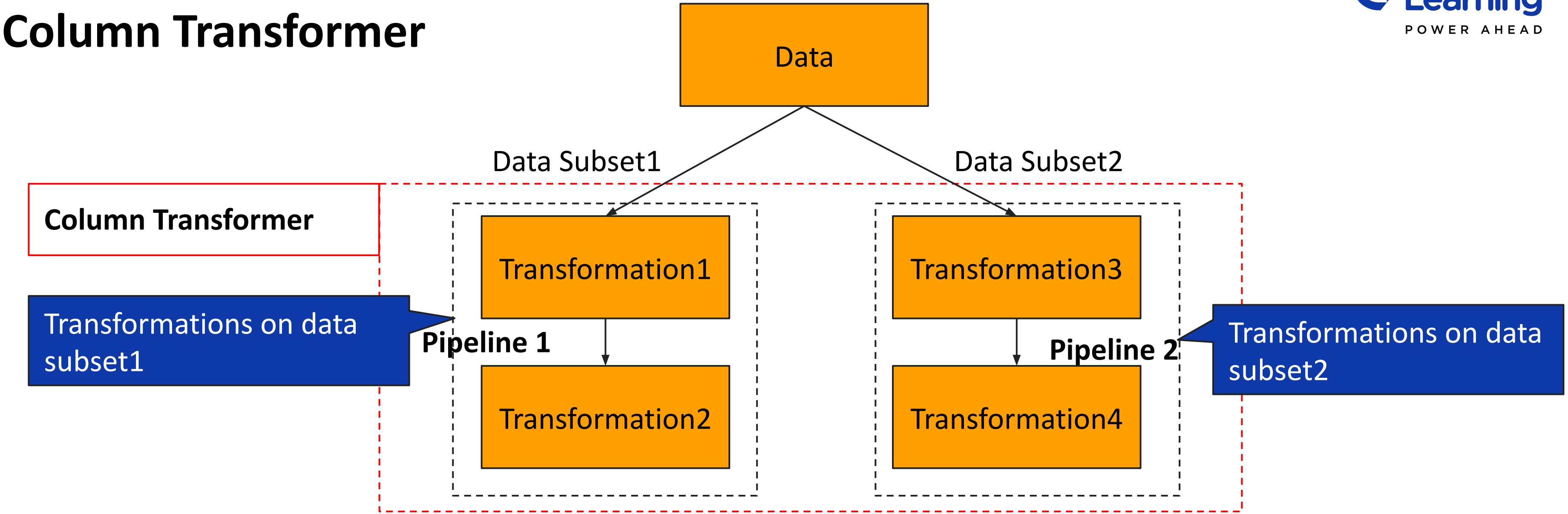
Column Transformer

- Pipelines helps to standardize the model building, but the steps in a pipeline are applied to each and every column
- We require more personalization and not just a single pipeline so that we are able to perform different processing steps on different columns/subsets of columns
- Column transformer allows different columns or column subsets of the input to be transformed separately, and the features generated by each transformer will be concatenated to form a single feature space.
- Column transformer is useful for heterogeneous or columnar data, and allows us to combine several feature extraction mechanisms or transformations into a single transformer

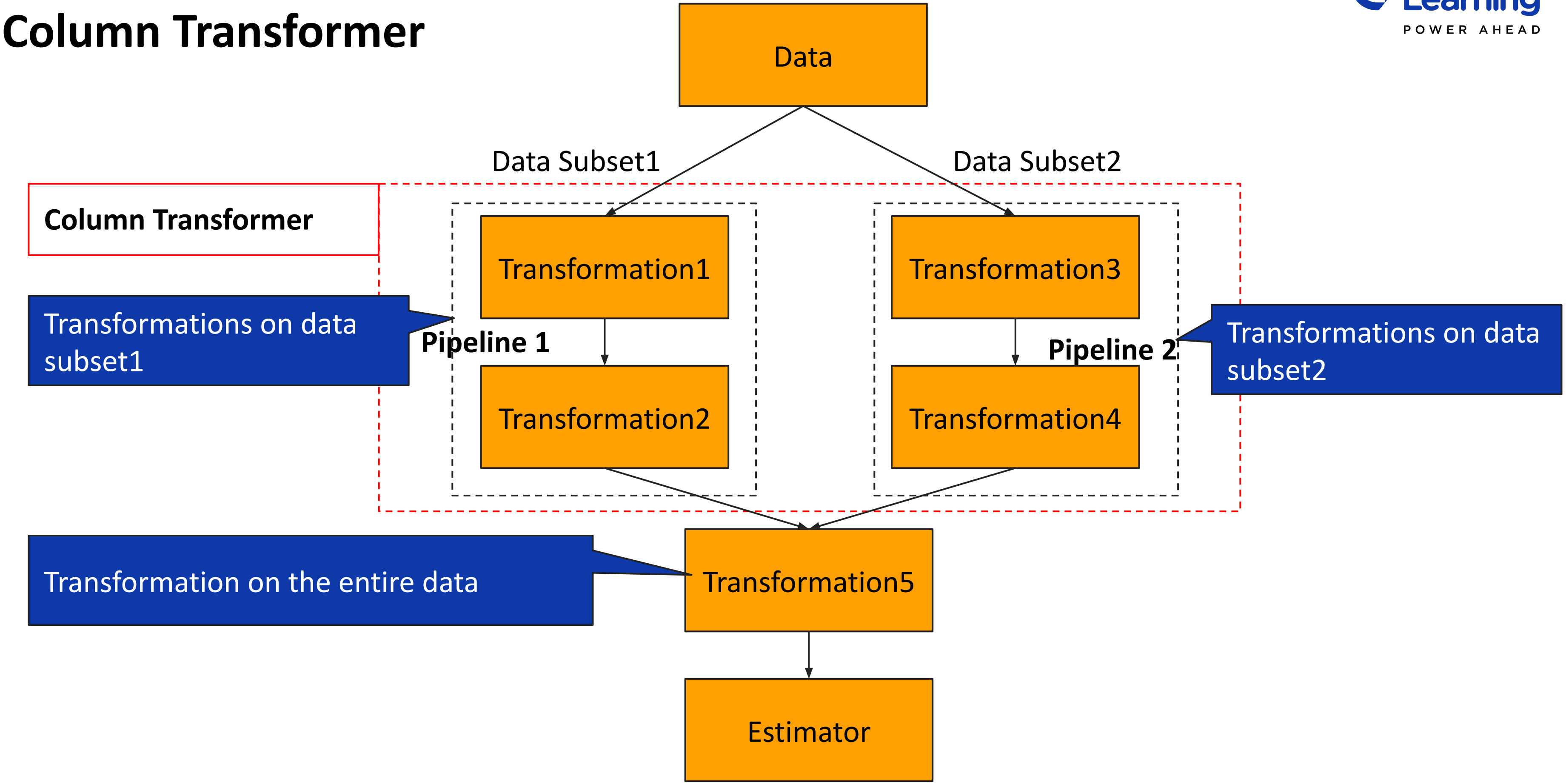
Column Transformer



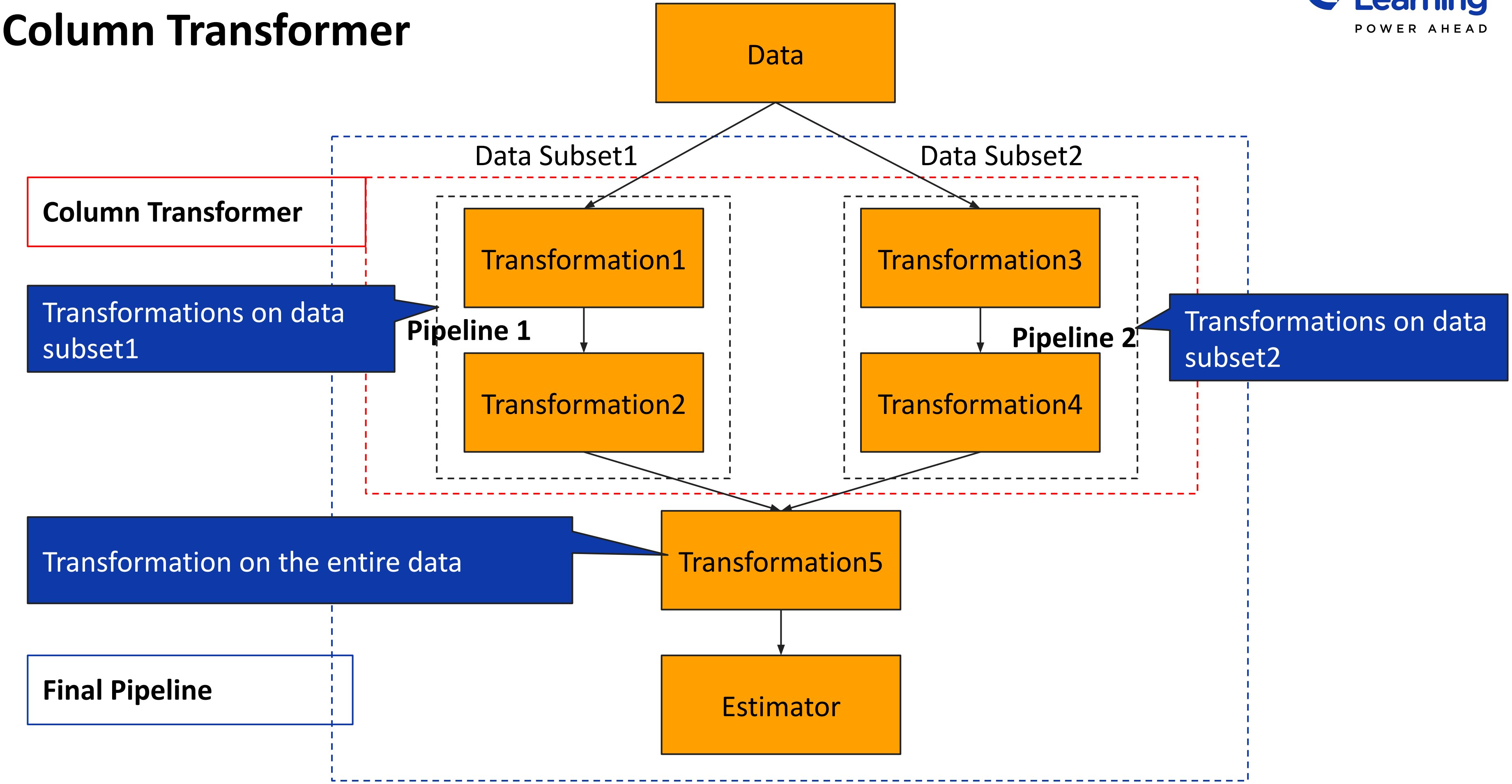
Column Transformer



Column Transformer



Column Transformer



greatlearning

Power Ahead

Happy Learning !