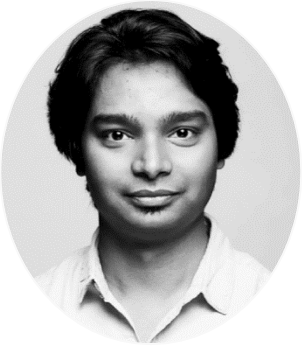
Preparing Data for Machine Learning Model: Part 2

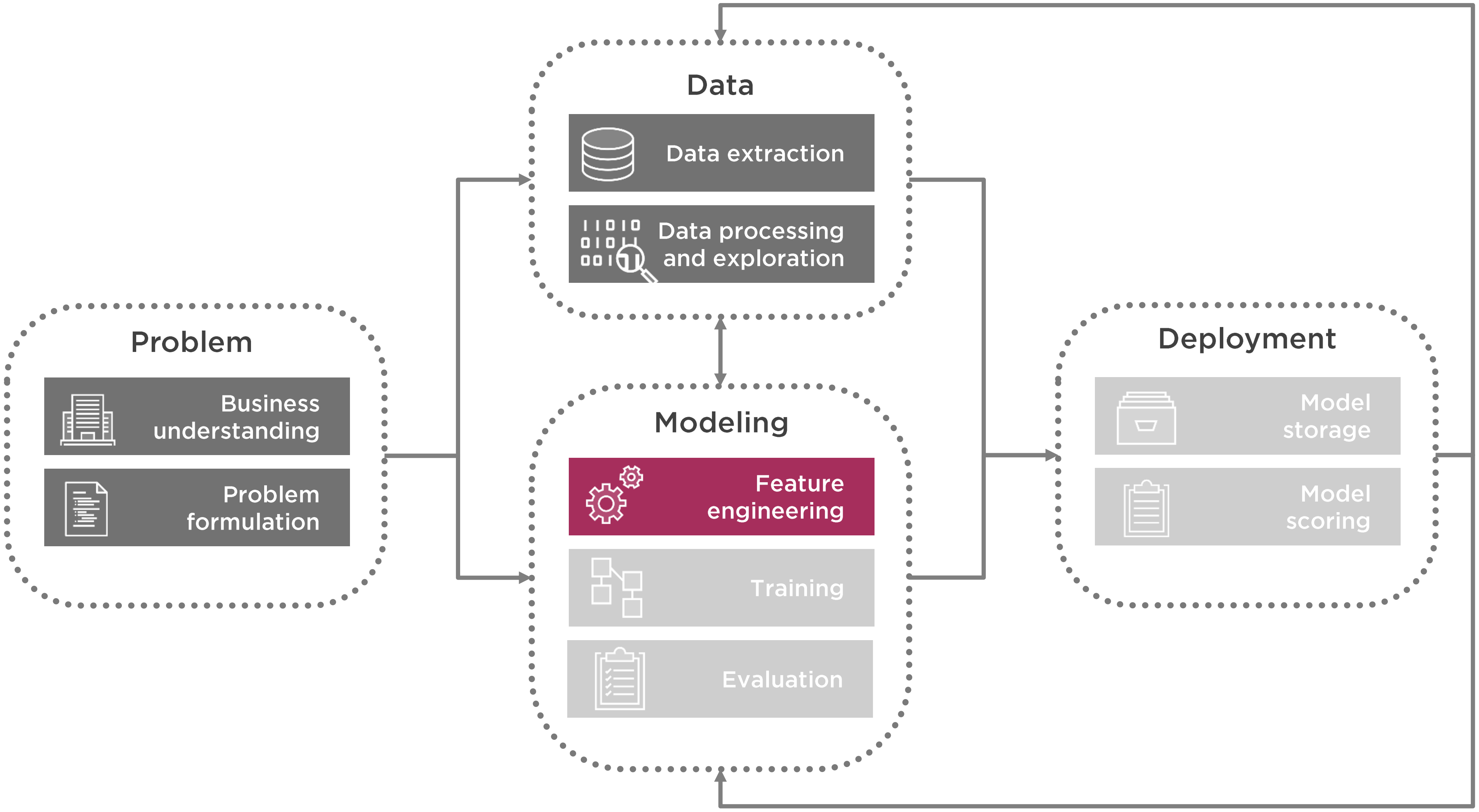


Abhishek Kumar

DATA SCIENTIST | AUTHOR | SPEAKER

@meabhishekkumar

Machine Learning Workflow



Overview

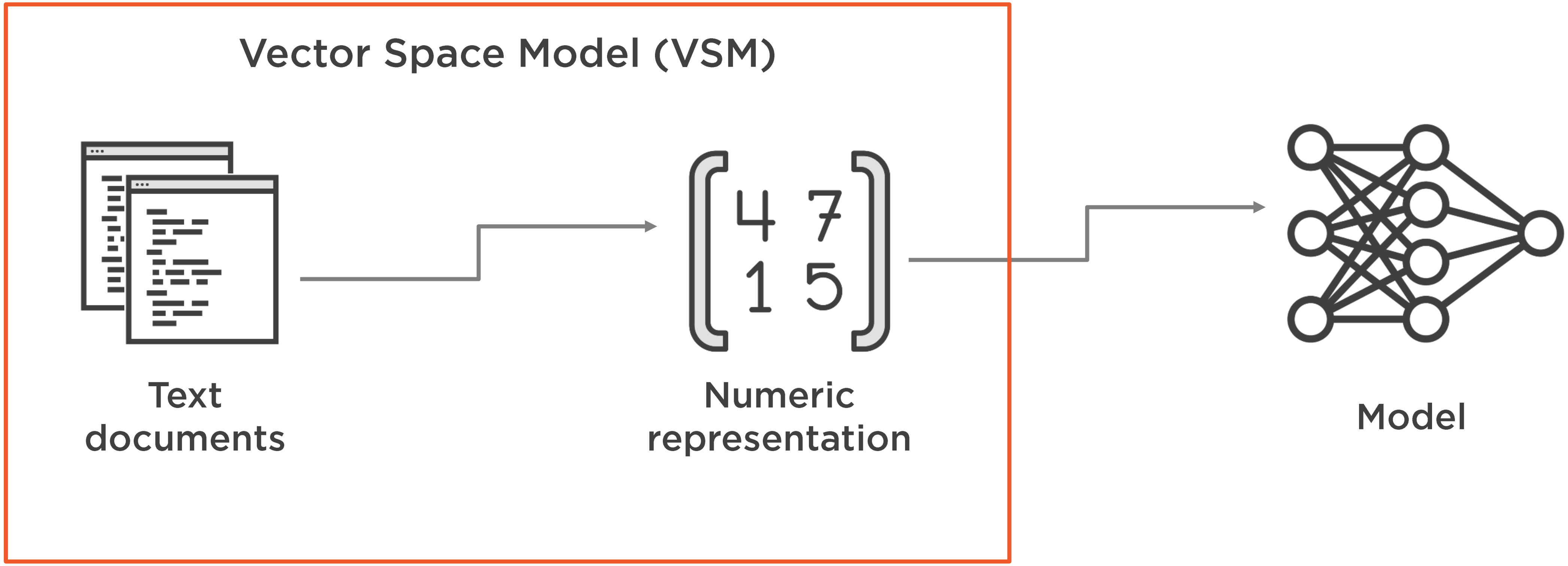
Generate features from text

Perform feature engineering

- Map approach
- Function generator approach

Prepare dataset for train and test

Generating Features from Text



Vector Space Model (VSM)

Text documents

Numeric representation

Model

Vector Space Model

	Step 1: Document vector		Step 2
I loved the movie	[I, loved, the, movie]	[loved, movie]	$\begin{bmatrix} 4 & 7 \\ 1 & 5 \end{bmatrix}$
Movie was boring	[movie, was, boring]	[movie, boring]	

Term Frequency(TF)

Step 1: Document vector

[I, loved, the, movie]

[loved, movie]

[movie, was, boring]

[movie, boring]

Step 2

Total documents count = N = 2
Dictionary size = 3

	Movie	Loved	Boring
Doc 1	1	1	0
Doc 2	1	0	1

Term document matrix

Term frequency (TF)

Inverse Document Frequency(IDF)

Step 2

Total documents count = N = 2

	Movie	Loved	Boring
Doc 1	1	1	0
Doc 2	1	0	1

Term frequency (TF)

	Movie	Loved	Boring
DF	2	1	1
IDF	1	1.693	1.693

Document frequency (DF)

Inverse document frequency (IDF) = log (N/DF) + 1

TFIDF

Step 2

Total documents count = N = 2

	Movie	Loved	Boring
Doc 1	1	1	0
Doc 2	1	0	1

Term frequency (TF)

	Movie	Loved	Boring
IDF	1	1.693	1.693

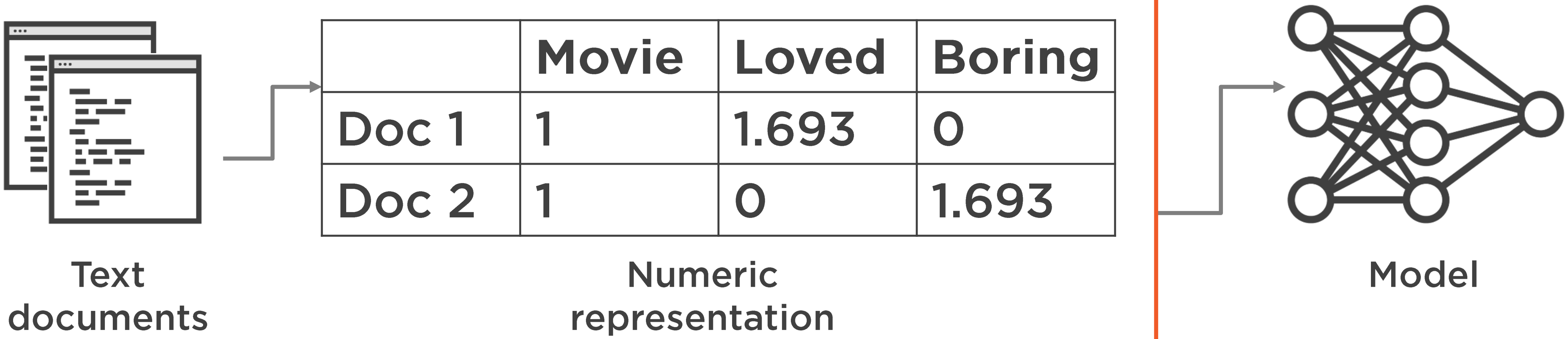
Inverse document frequency (IDF)

	Movie	Loved	Boring
Doc 1	1	1.693	0
Doc 2	1	0	1.693

TFIDF = TF*IDF

Generating Features from Text

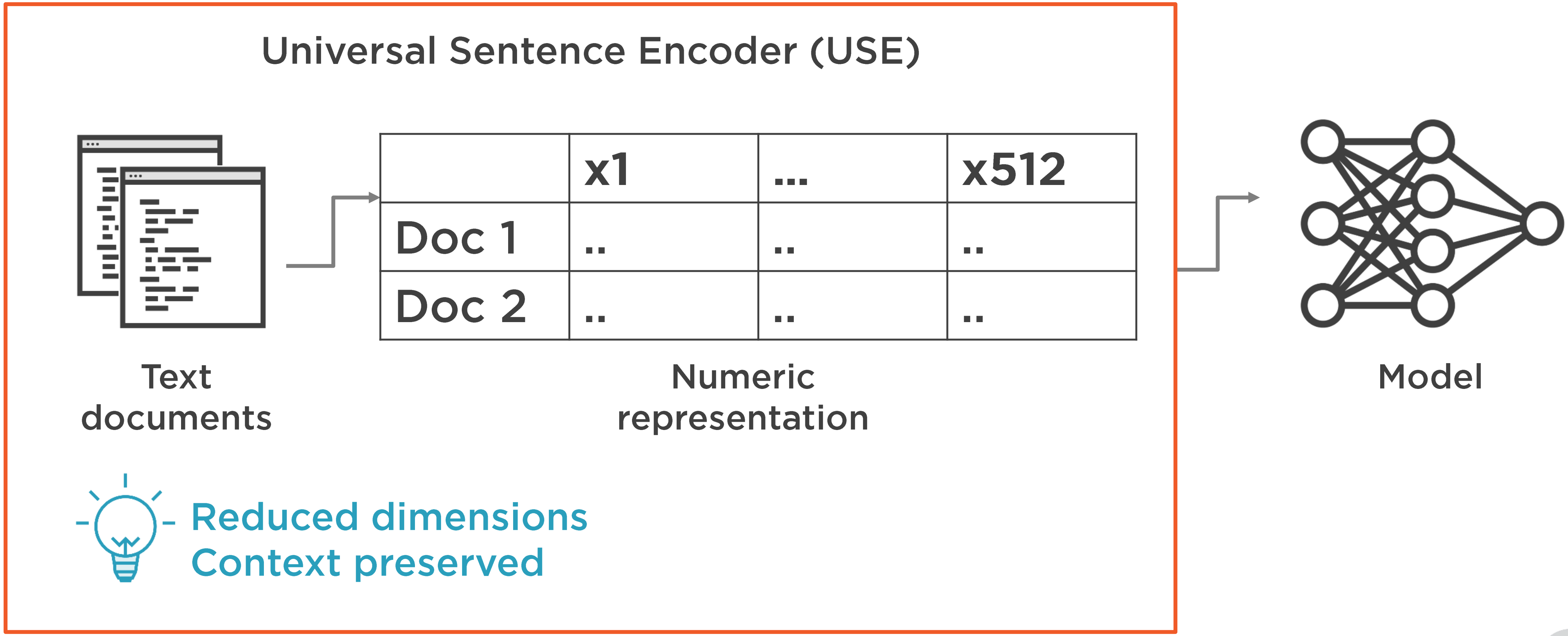
Vector Space Model (VSM)



	Movie	Loved	Boring
Doc 1	1	1.693	0
Doc 2	1	0	1.693

Dictionary size (dimensions) can become huge
Context not preserved

Generating Features from Text



Universal Sentence Encoder (USE)

	x1	...	x512
Doc 1	..	..	..
Doc 2	..	..	..

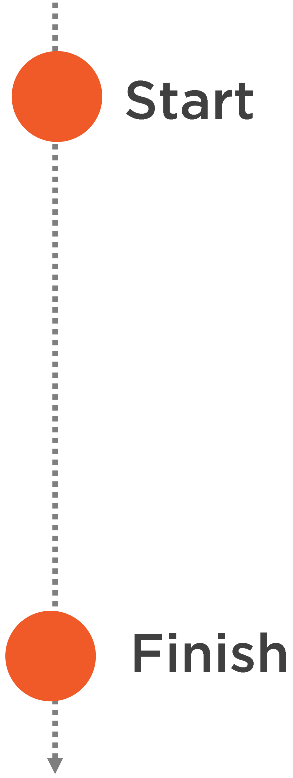
Reduced dimensions
Context preserved

Demo

Creating TFIDF features

Function Generator

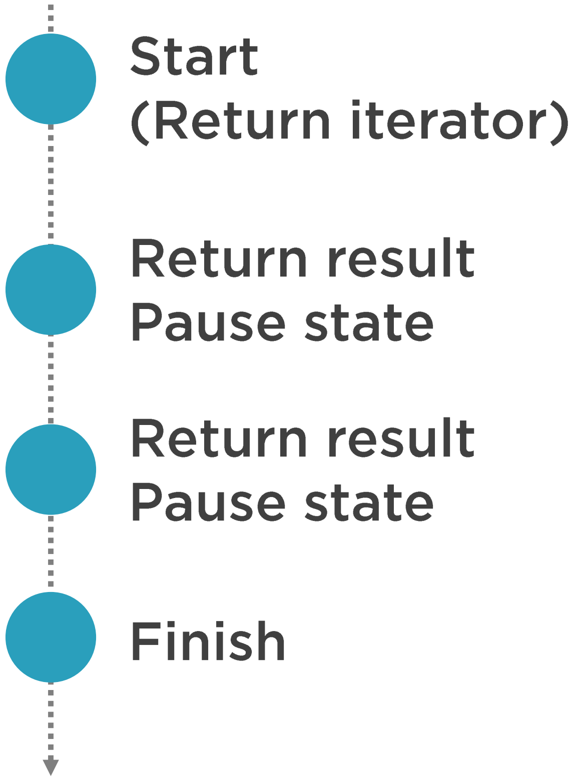
```
const func1 = () =>
{

// do A

// do B



}
```



Start

Finish

```
function* func2Gen()
{


yield


yield


}
```



Start
(Return iterator)

Return result
Pause state

Return result
Pause state

Finish

Function Generator

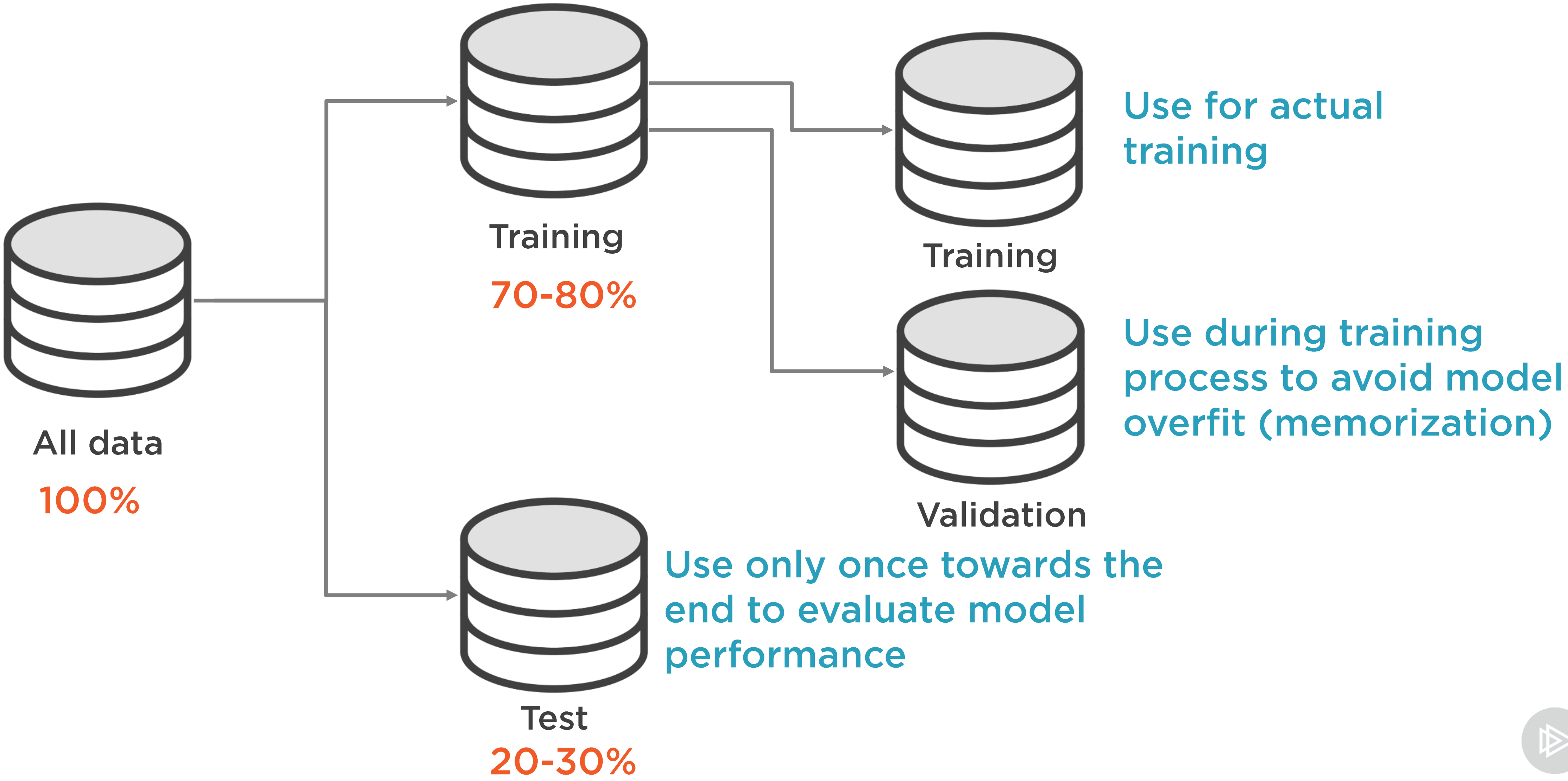
```
function* readFiles() {

filePaths = getAllFilePath();

for(let i=0; i < filePaths.length ; i ++) {

     data = readFile(filePaths[i]);
     yield data;
   }

}
```

- Useful to read or process data in chunks
- Reduce memory issues

Demo

Creating feature dataset using generators

Train Validation Test Split



All data

100%

Training

70-80%

Training

Use for actual training

Validation

Use during training process to avoid model overfit (memorization)

Test

20-30%

Use only once towards the end to evaluate model performance

Demo

Splitting data into train, validation, and test datasets

Summary

Feature engineering

- Vector space model

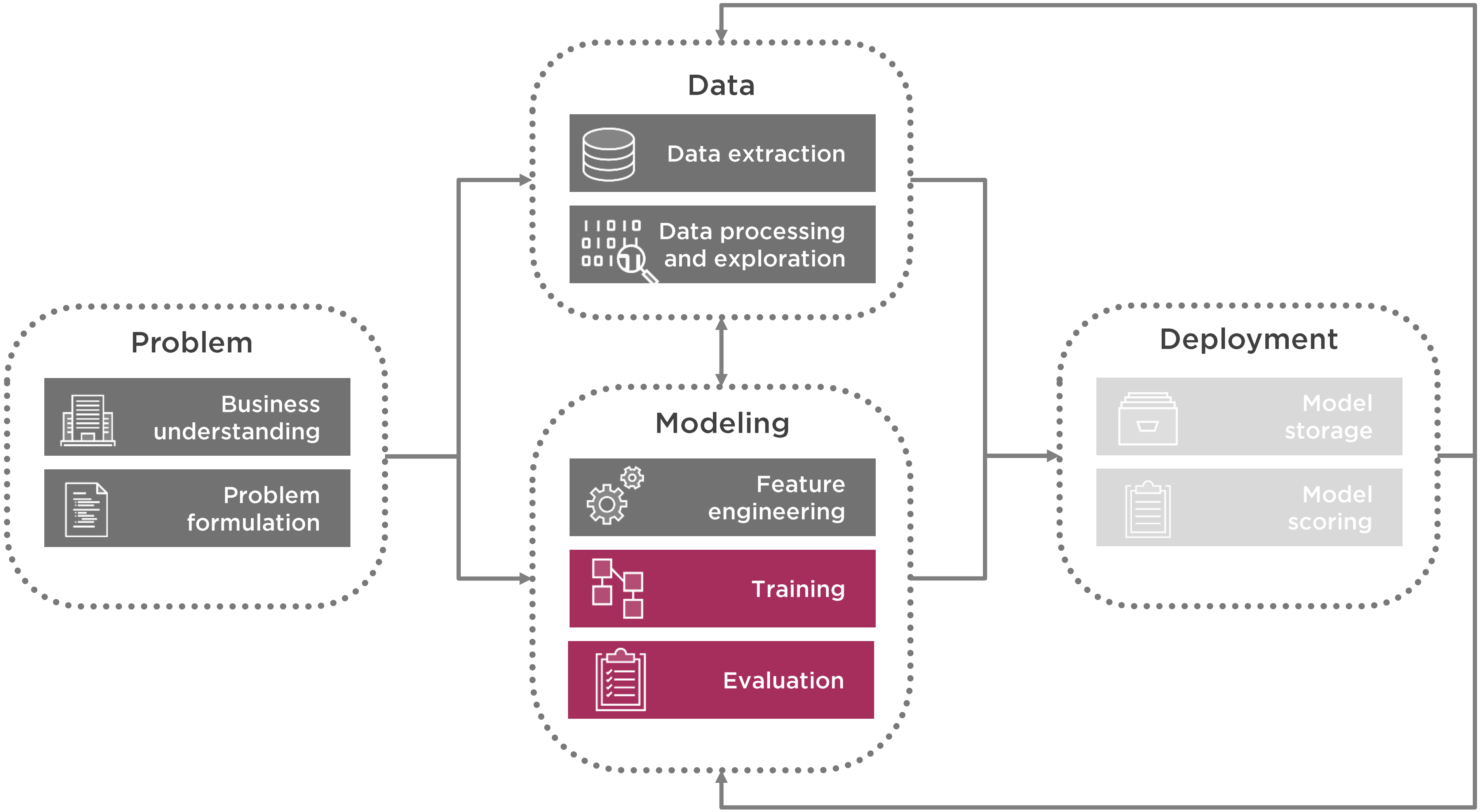
Map approach

Function generator approach

Train, validation, and test split

- Shuffle, take, skip, and batch

Machine Learning Workflow



Up Next: Building, Training, and Evaluating Machine Learning Model