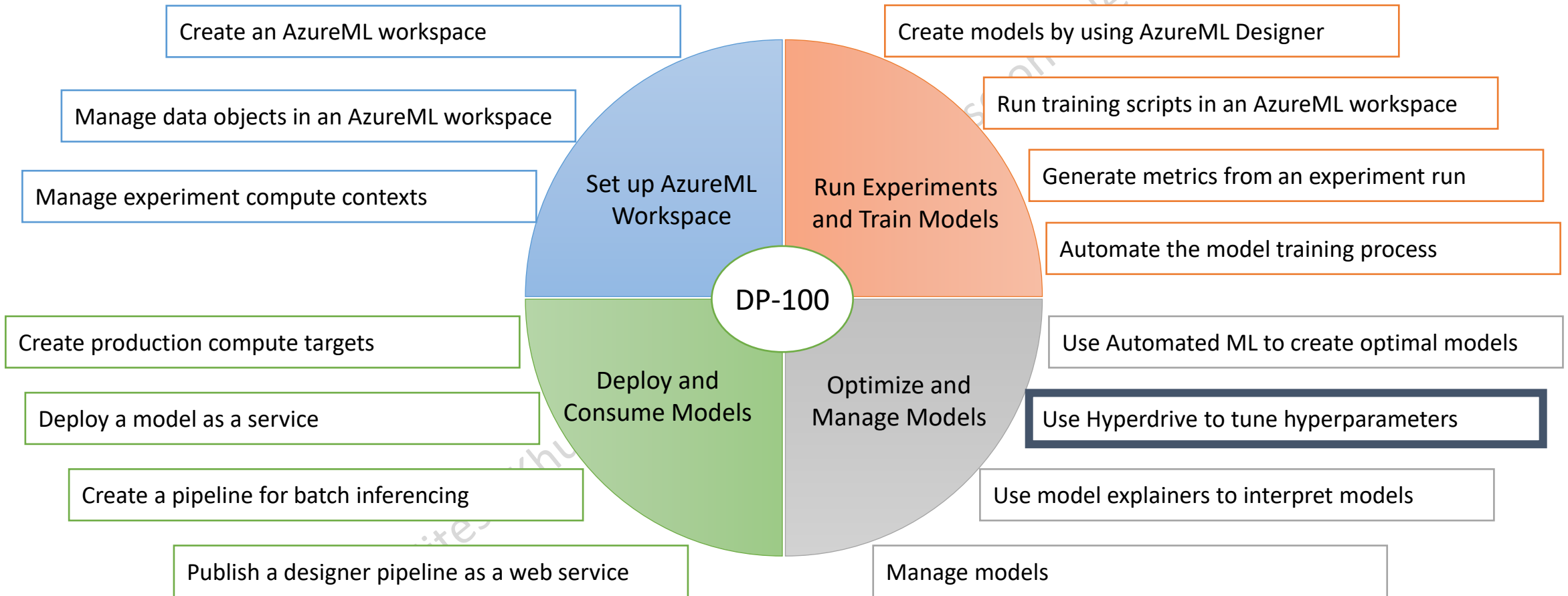


Azure Machine Learning



Models and Parameters



Two-Class Logistic Regression

1

Create trainer mode

Single Parameter ▼

Optimization tolerance

1E-07

L1 regularization weight

1

L2 regularization weight

1

Memory size for L-BFGS

20

Random number seed

☒ Allow unknown categorical levels



Two-Class Boosted Decision...

1

Create trainer mode

Single Parameter ▼

Maximum number of leaves per tree

20

Minimum number of samples per leaf node

10

Learning rate

0.2

Number of trees constructed

100

Random number seed

☒ Allow unknown categorical levels



Two-Class Support Vector M...

1

Create trainer mode

Single Parameter ▼

Number of iterations

1

Lambda

0.001

☒ Normalize features

☐ Project to the unit-sphere

Random number seed

☒ Allow unknown categorical levels

Tune Model Hyperparameters

- Helps in determining the best possible combination of hyperparameters
- Also known as hyperparameter optimization
- Performance metric to measure
 - Accuracy
 - Precision
 - Recall
 - AUC
 - F1Score

Designer

Tune Model Hyperparameters ✕

Specify parameter sweeping mode ⓘ *

Random sweep ▼

Maximum number of runs on random sweep...

5

Random seed ⓘ *

0

Label column ⓘ * [Edit column](#)

A value is required.

Metric for measuring performance for classification...

Accuracy ▼

Metric for measuring performance for regression ⓘ *

Mean absolute error ▼

Classic Studio

Draft saved at 9:29:24 AM

Tune Model Hyperparameters ⓘ

Specify parameter sweeping mode

Random sweep ▼

Maximum number of runs on r... ⓘ

5

Random seed ⓘ ⓘ

0

Label column

Selected columns:
Launch the selector tool to make a selection

[Launch column selector](#)

Metric for measuring performa... ⓘ

Accuracy ▼

Metric for measuring performa... ⓘ

Mean absolute error ▼

Parameter Search Space

Discrete

- Discrete with finite set of values
- [4, 5, 8, 15, 20]



Two-Class Boosted Decision...

1

Create trainer mode

Single Parameter

Maximum number of leaves per tree

20

Minimum number of samples per leaf node

10

Learning rate

0.2

Number of trees constructed

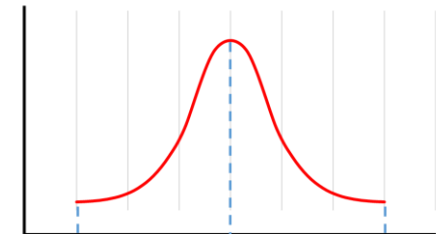
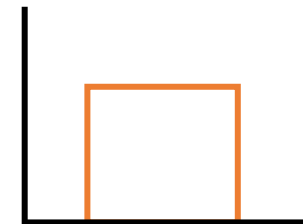
100

Random number seed

☒ Allow unknown categorical levels

Continuous

- Specified as a distribution over a continuous range
- Uniform (low, high)
- Log-Uniform (low, high)
- Normal (mu, stddev)
- Log-Normal (mu, stddev)



Select a Sampling Method

- Random Sampling
- Grid Sampling
- Bayesian Sampling

© Jitesh Khurkhuriya – AzureML Course on Udemy

What is a Grid?

- Cartesian Product of Parameters
- Parameter 1 → 1, 2, 3
- Parameter 2 → A, B, C, D

← Parameter 2

Parameter 1 →

	1	2	3
A	A, 1	A, 2	A, 3
B	B, 1	B, 2	B, 3
C	C, 1	C, 2	C, 3
D	D, 1	D, 2	D, 3

Random Sampling

← Parameter 2

Parameter 1 →

	1	2	3
A	A, 1	A, 2	A, 3
B	B, 1	B, 2	B, 3
C	C, 1	C, 2	C, 3
D	D, 1	D, 2	D, 3

Grid Sampling

← Parameter 2

Parameter 1 →

	1	2	3
A	A, 1	A, 2	A, 3
B	B, 1	B, 2	B, 3
C	C, 1	C, 2	C, 3
D	D, 1	D, 2	D, 3

Sampling Method Comparison

Random Sampling	Grid Sampling	Bayesian Sampling
<ul style="list-style-type: none">• Supports Discrete as well as Continuous Hyperparameters• Selects the combination with best results• Supports early termination	<ul style="list-style-type: none">• Supports only Discrete Hyperparameters• Selects the combination with best results• Supports early termination	<ul style="list-style-type: none">• Support Discrete as well as Continuous Hyperparameters• Selects the combination by learning from the previous run• Uses Bayesian Optimization for getting the best combination.• Does not support early termination

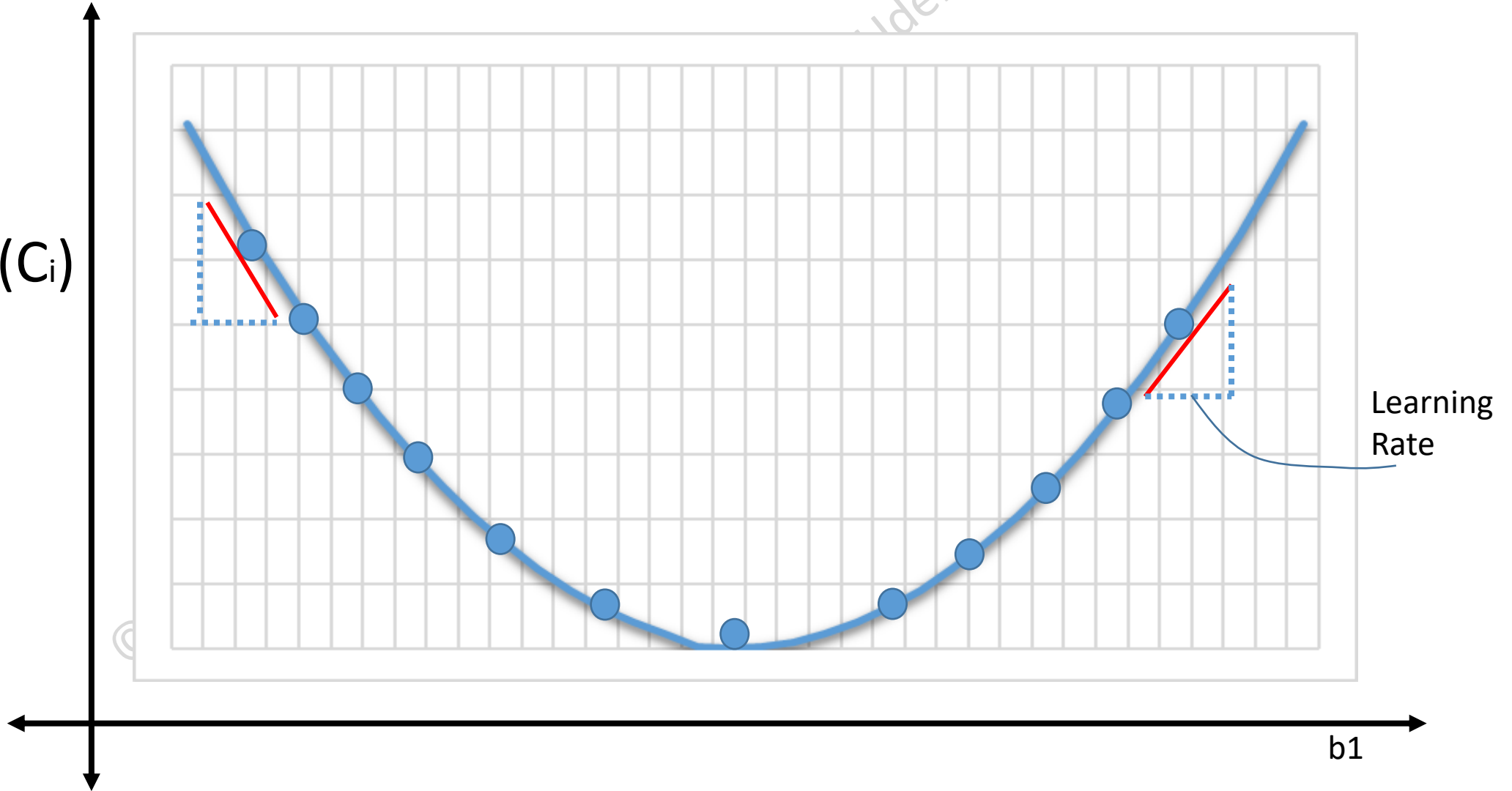
What is Early Termination Policy?

© Jitesh Khurkhuriya – AzureML Course on Udemy

Gradient Descent

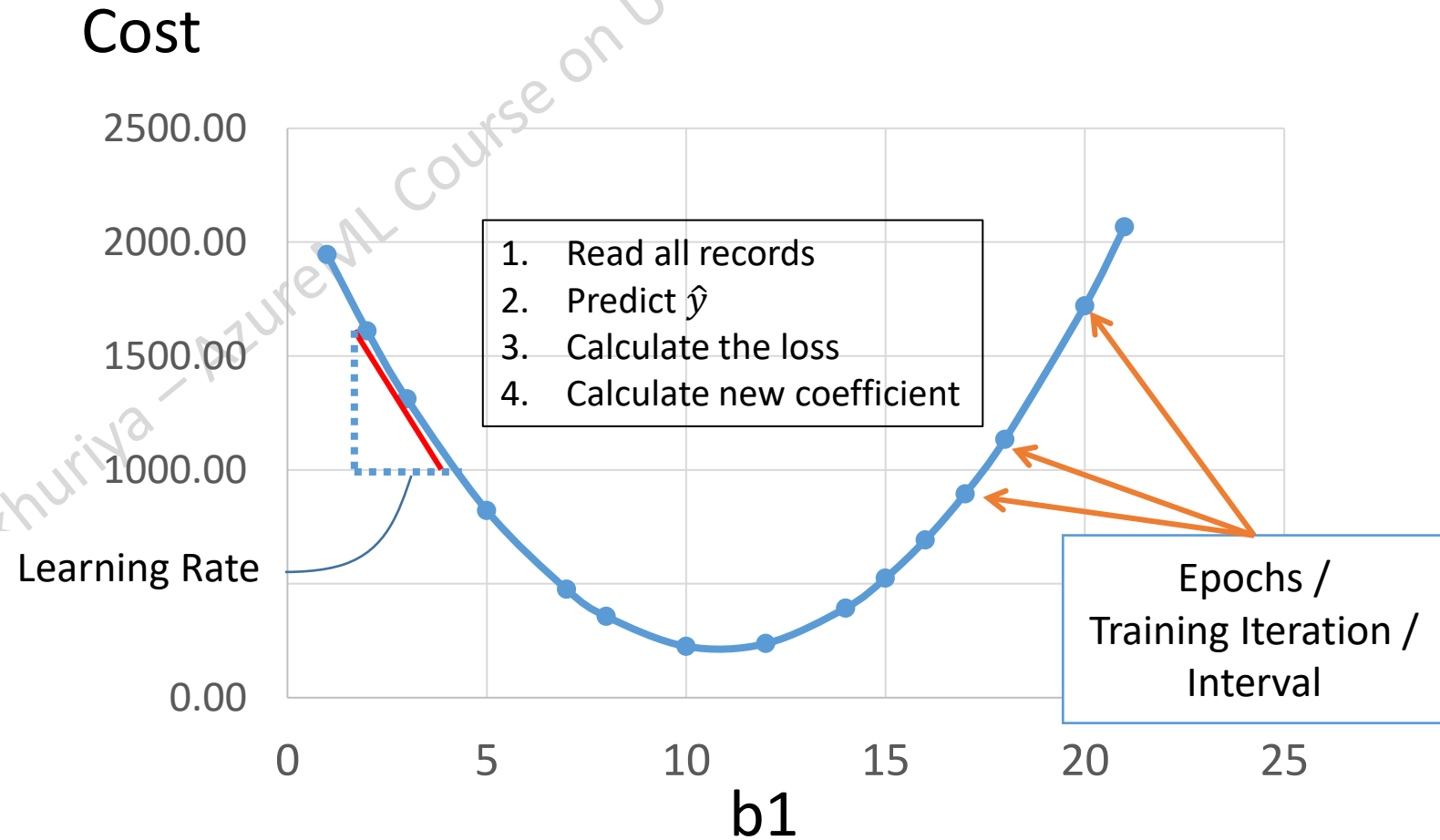
Cost Function: $C(b_1)$

$$b_j := b_j - \alpha f(C_i)$$

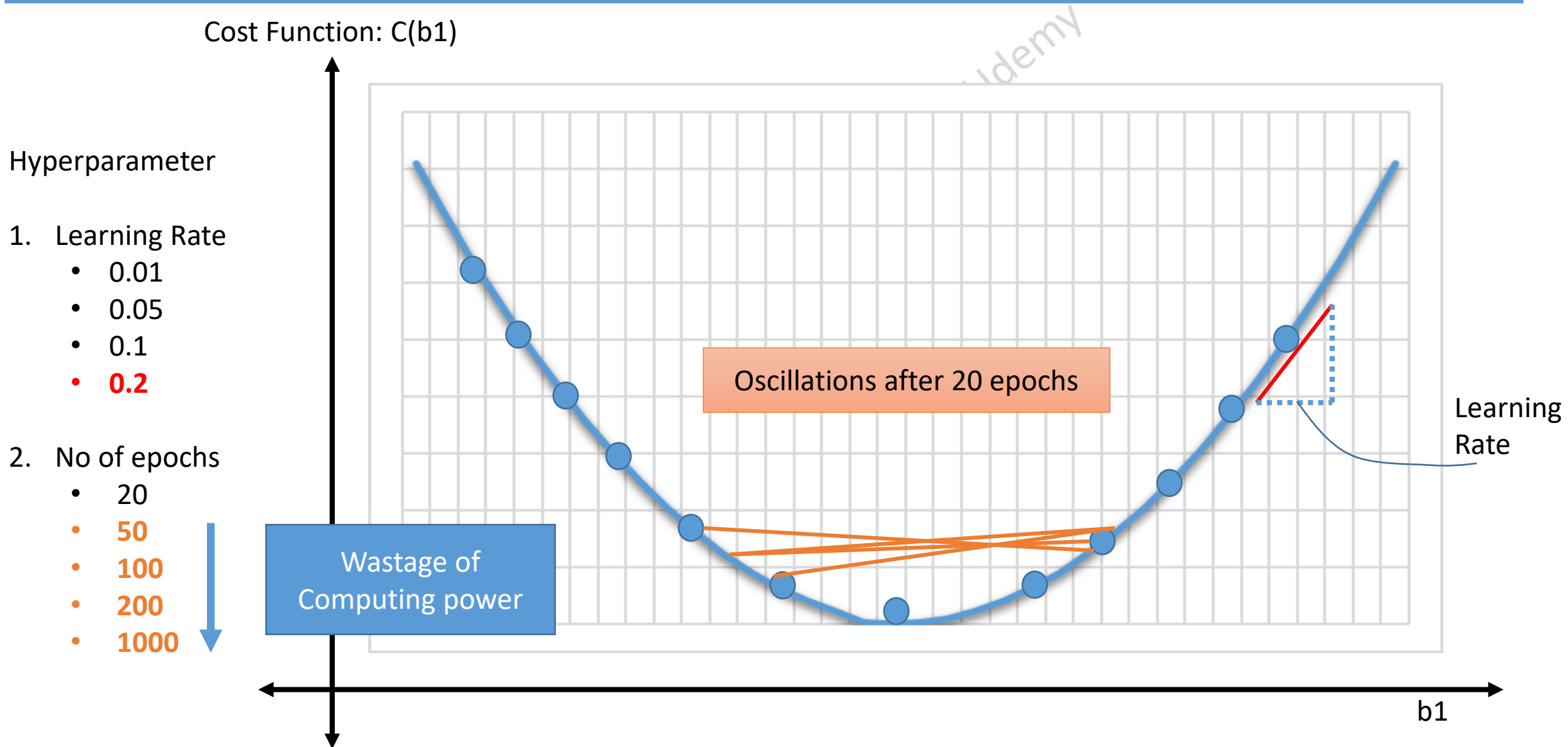


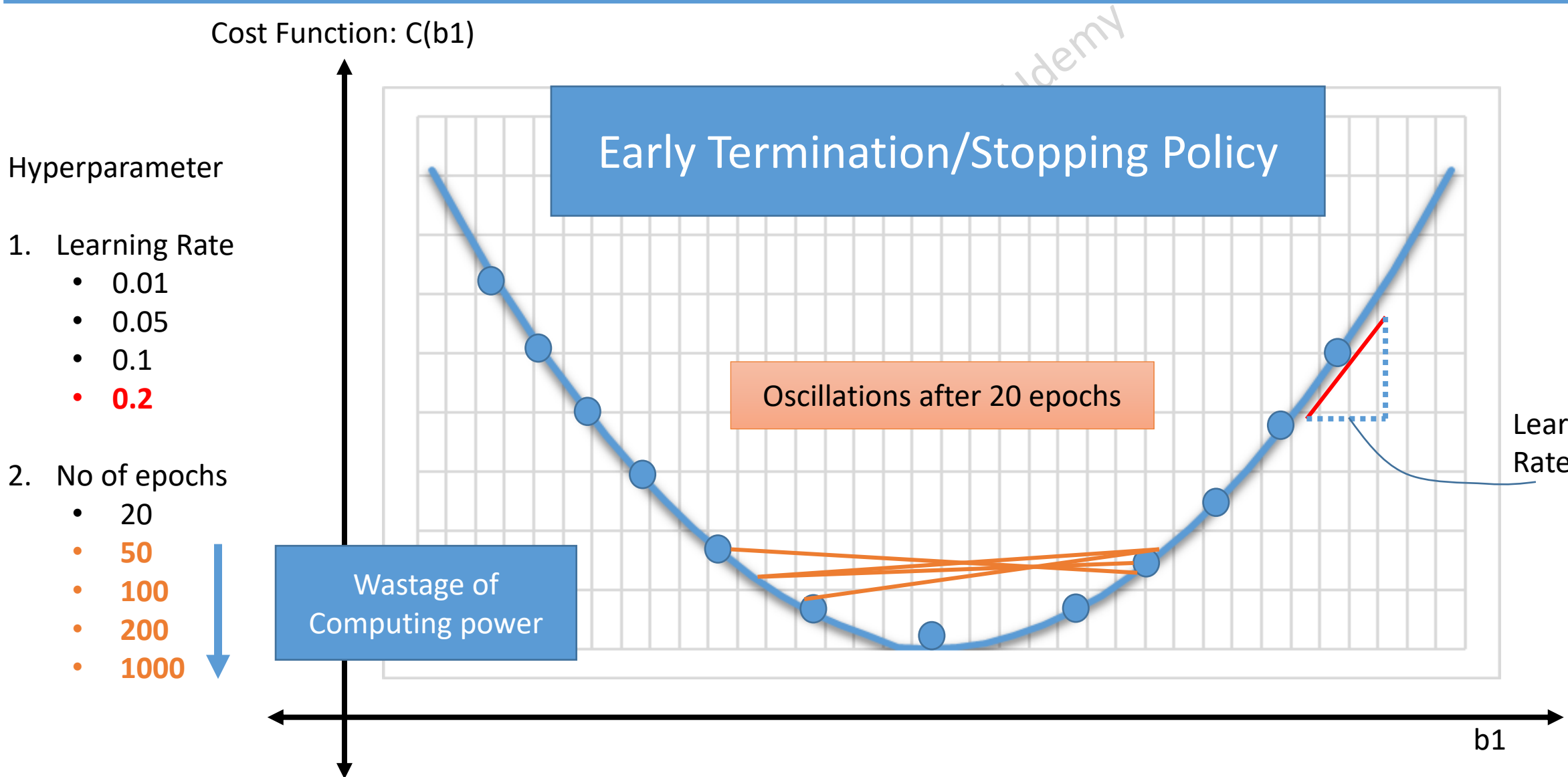
Cost Function Plot

b0	b1	cost
0	1	1944.54
0	2	1610.08
0	3	1311.46
0	5	821.77
0	7	475.46
0	8	356.08
0	10	224.85
0	12	237.00
0	14	392.54
0	15	524.08
0	16	691.46
0	17	894.69
0	18	1133.77
0	20	1719.46
0	21	2066.08



Gradient Descent





Types of Early Termination Policies in Hyperdrive

- Bandit Policy
- Median Stopping Policy
- Truncation Selection Policy
- No Termination Policy

© Jitesh Khurkhuriya – AzureML Course on Udemy

Bandit Policy – Slack Amount

```
from azureml.train.hyperdrive import BanditPolicy

early_termination_policy = BanditPolicy(slack_amount = 0.1,
                                       evaluation_interval=1,
                                       delay_evaluation=10)
```

- Slack Factor
 - Apply the policy after delay_evaluation of 10 epochs.
 - After the delay of 10, apply the policy on every epoch.
 - Check if the primary metric of the new epoch is less than, (PM of Best Epoch – 0.1) then terminate/stop the Run
- Slack Amount
- Evaluation interval
- Delay Evaluation

e.g.

During the first 10 epochs, if epoch 5 provides the best accuracy of 0.90, then if any epoch after 10th one, reports accuracy less than (0.90 – 0.10 = 0.80), the run will be terminated/stopped.

Bandit Policy – Slack Factor

```
from azureml.train.hyperdrive import BanditPolicy

early_termination_policy = BanditPolicy(slack_factor = 0.1,
                                       evaluation_interval=1,
                                       delay_evaluation=10)
```

- Slack Factor
 - Apply the policy after delay_evaluation of 10 epochs.
 - After the delay of 10, apply the policy on every epoch.
 - Check if the best primary metric is greater than,
(Current PM + Current PM* 0.1) then terminate/stop the Run
- Slack Amount
- Evaluation interval
 - e.g.
 - During the first 10 epochs, if epoch 5 provides the best accuracy of 0.90. Then if the epoch 11th reports an accuracy of 0.8, it will compare ($0.8 + 0.8 \times 0.1 = 0.88$). As 0.9 is greater than 0.88, the run will be terminated/stopped.
- Delay Evaluation

Median Stopping Policy

```
from azureml.train.hyperdrive import MedianStoppingPolicy
```

```
early_termination_policy = MedianStoppingPolicy(evaluation_interval=1, delay_evaluation=5)
```

Epoch	Accuracy	Running average
1	0.85	0.850
2	0.86	0.855
3	0.87	0.860
4	0.87	0.862
5	0.88	0.866
6	??	

Median of all previous
Running Averages

Truncation Selection Policy

```
from azureml.train.hyperdrive import TruncationSelectionPolicy

early_termination_policy = TruncationSelectionPolicy(evaluation_interval=1,
                                                    truncation_percentage=20,
                                                    delay_evaluation=10)
```

Epoch	Accuracy
1	0.85
2	0.86
3	0.87
4	0.87
5	0.88
6	0.86
7	0.87
8	0.86
9	0.85
10	0.86

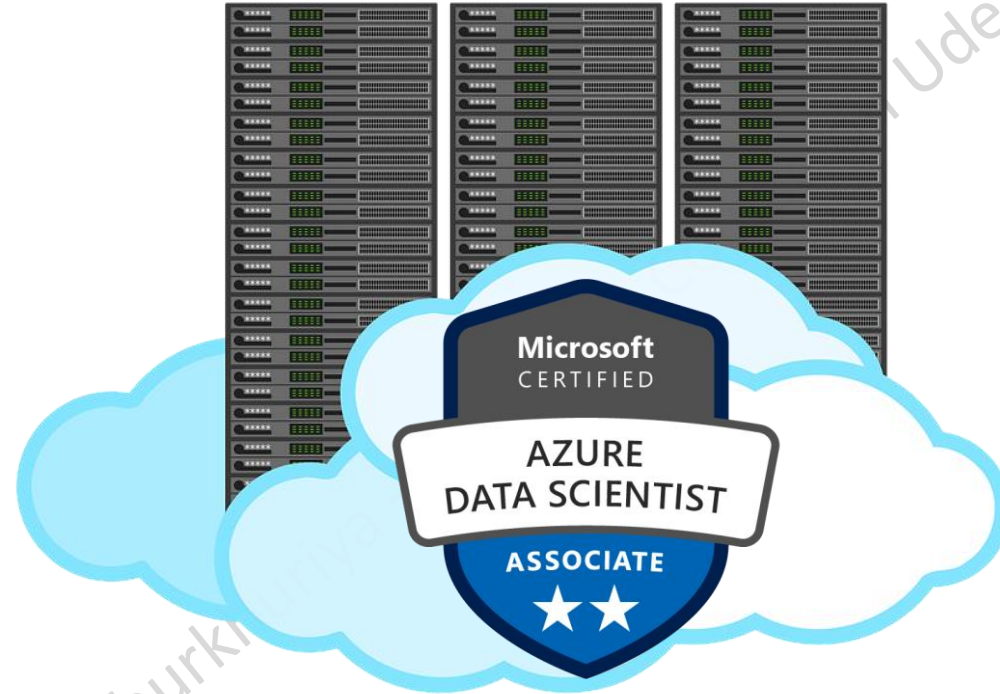
- Expunge the bottom 20% of the iterations
- Does not stop or terminate the Run.

When to apply Early Termination?

- Algorithms with learning rate and epochs
 - SGDClassifier
 - SGDRegressor
 - XGBoost
 - Deep learning



Azure Machine Learning



Thank You..!!