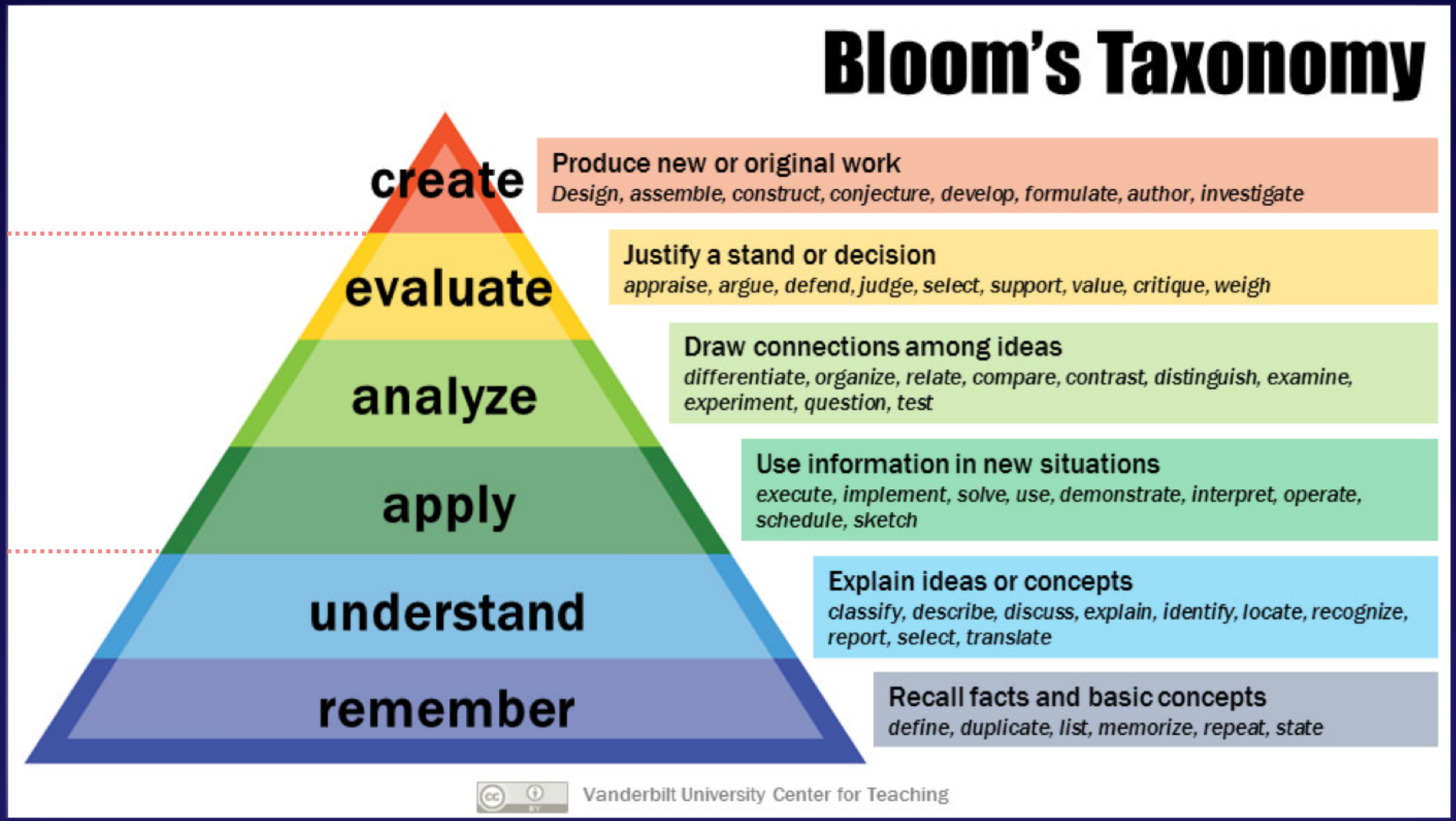
A CLOUD GURU

Data Preparation

Scott Pletcher
INSTRUCTOR

Bloom's Taxonomy

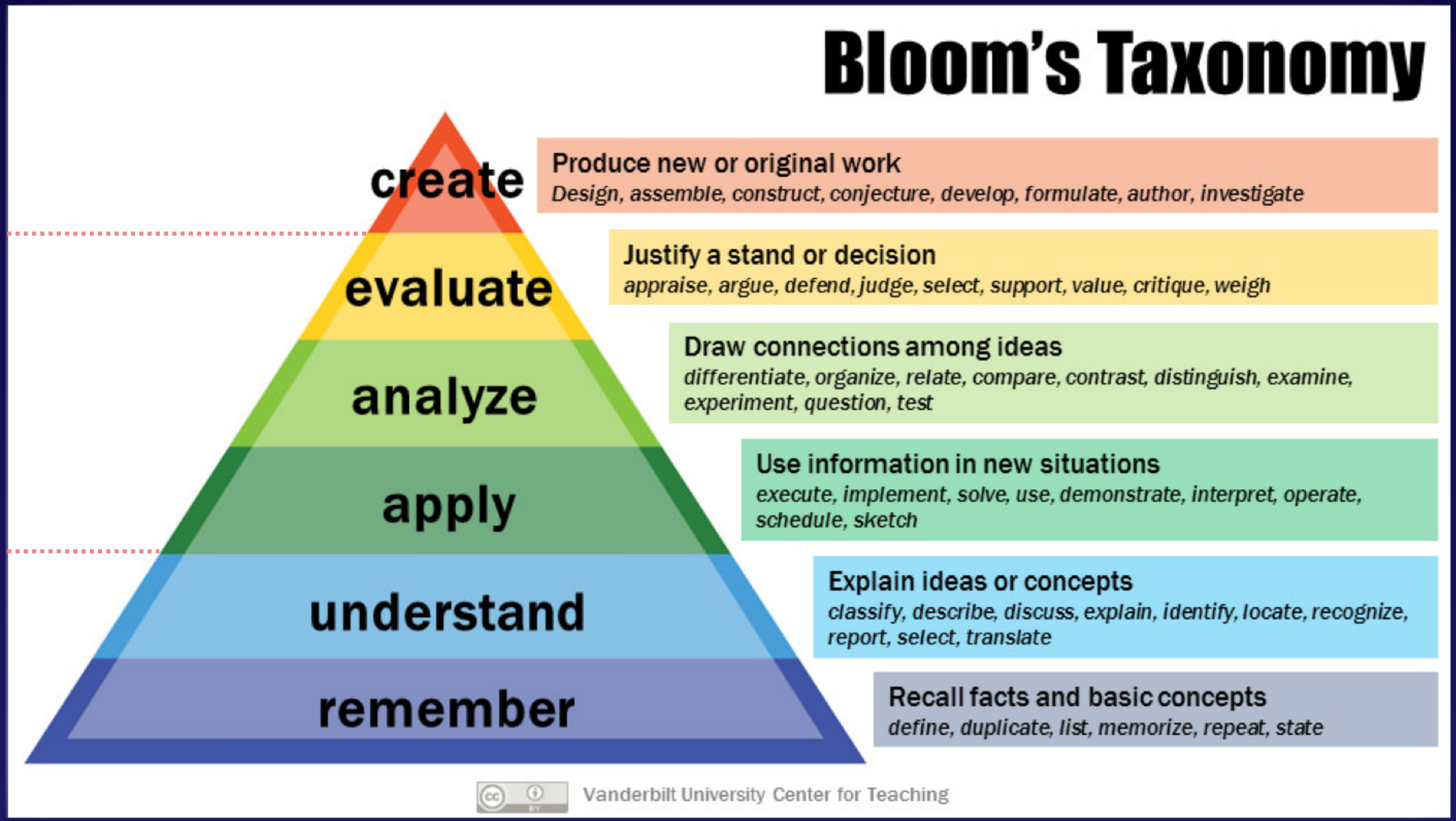


Bloom's Taxonomy

AWS Objective

Professional-Level Certification Objective

Associate-Level Certification Objective

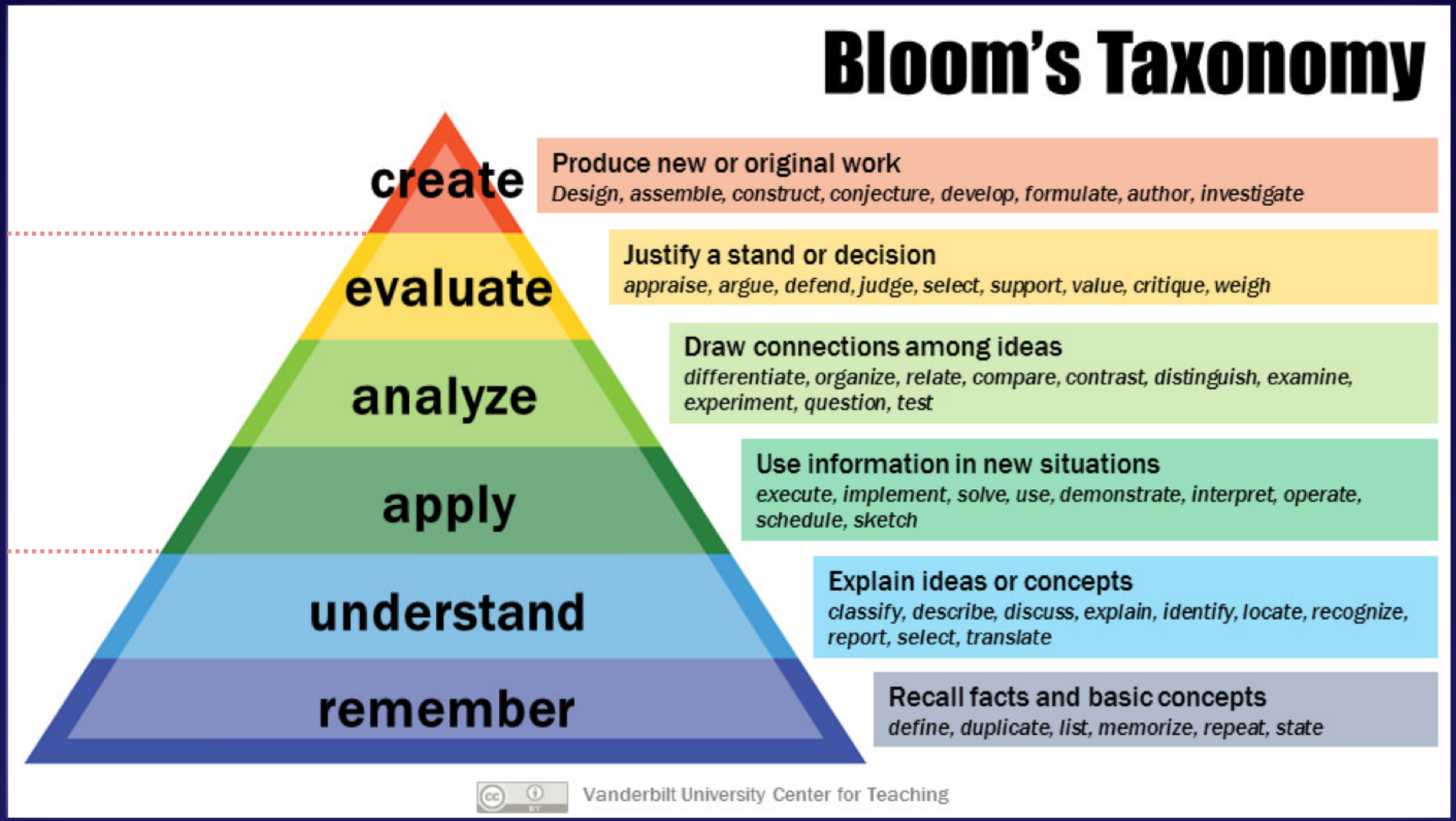


Bloom's Taxonomy

True Intelligence

Generalize

Memorize



ML Objective

We want
Generalization…

…not Memorization

Data Preparation

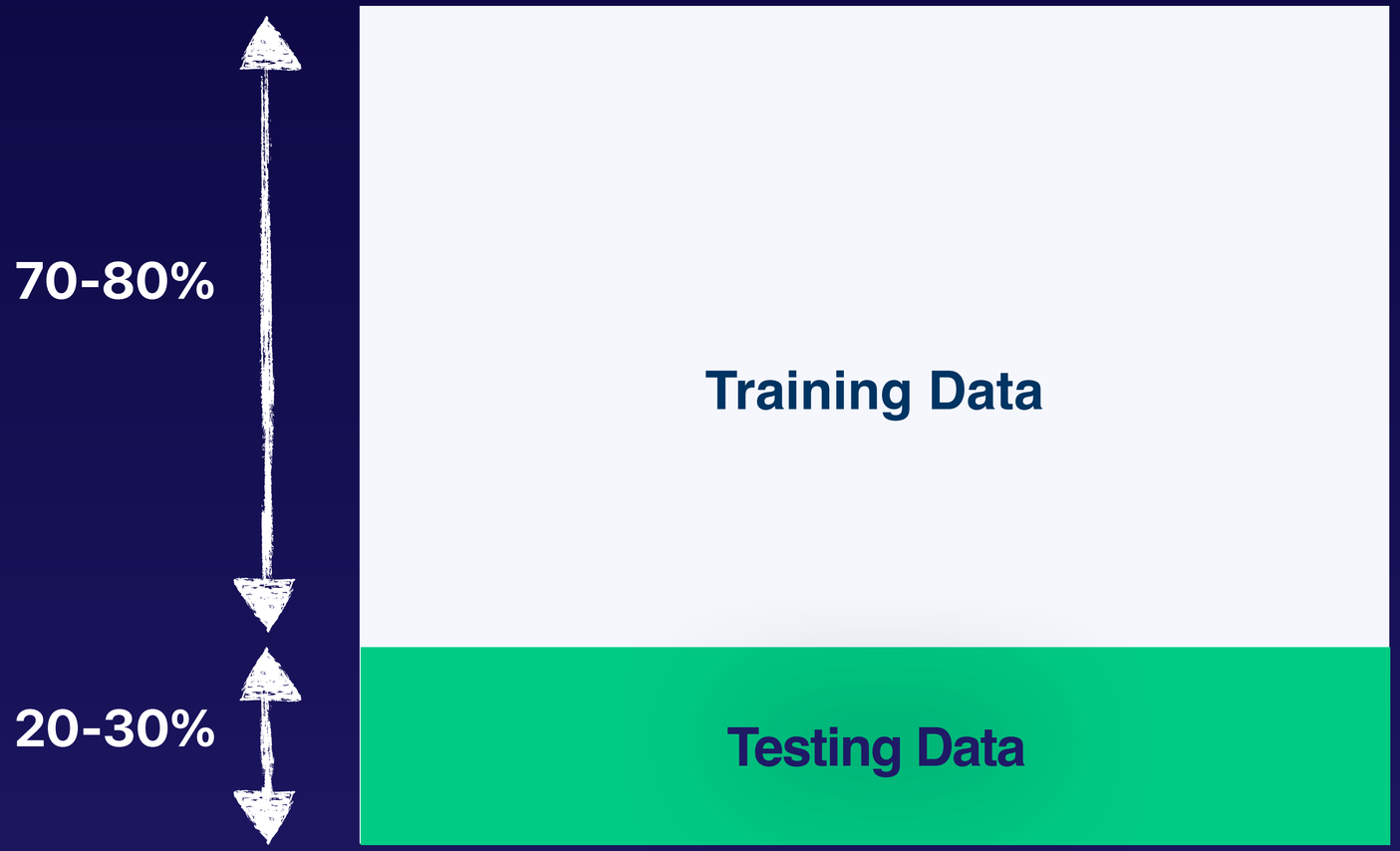
Use most data to train, but reserve some data to see if the model has really learned to generalize and not just repeating what we've already shown it.

Data Preparation

Our Data

Data Preparation

Our Data

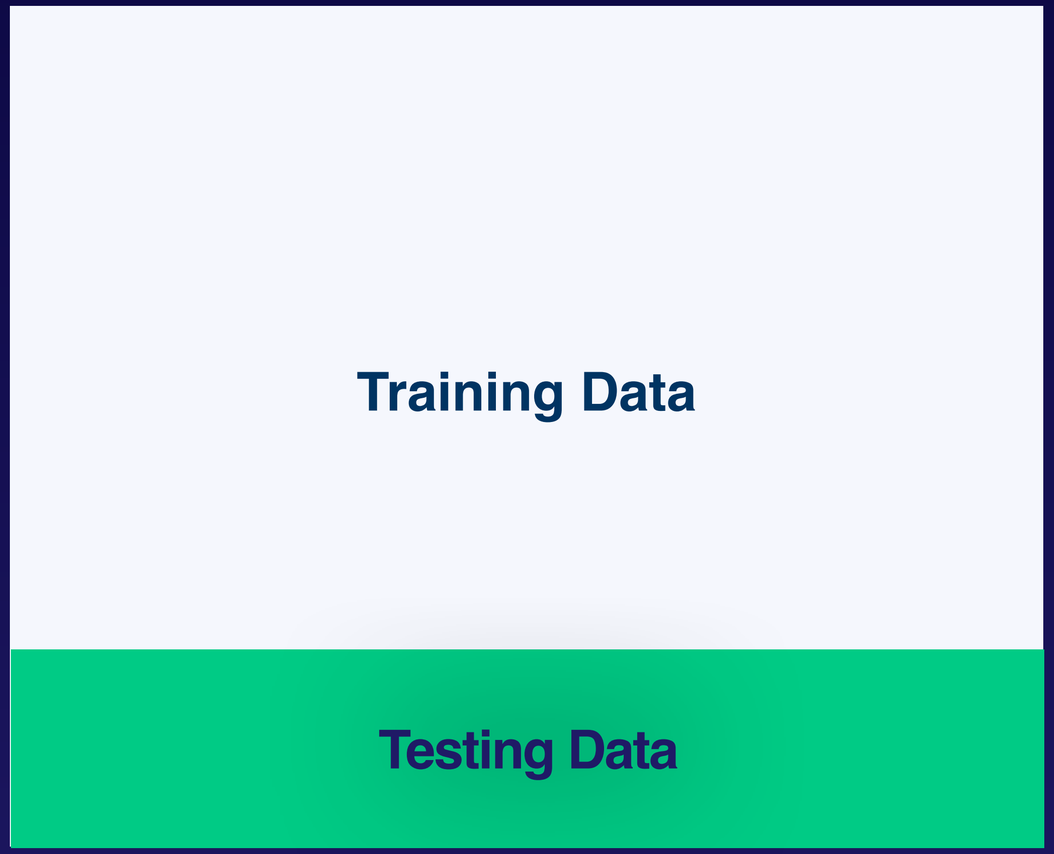


Data Preparation

Our Data

{ 0,2,0,3,0,2,4…}

{ 8,6,3,6,1,3,9…}

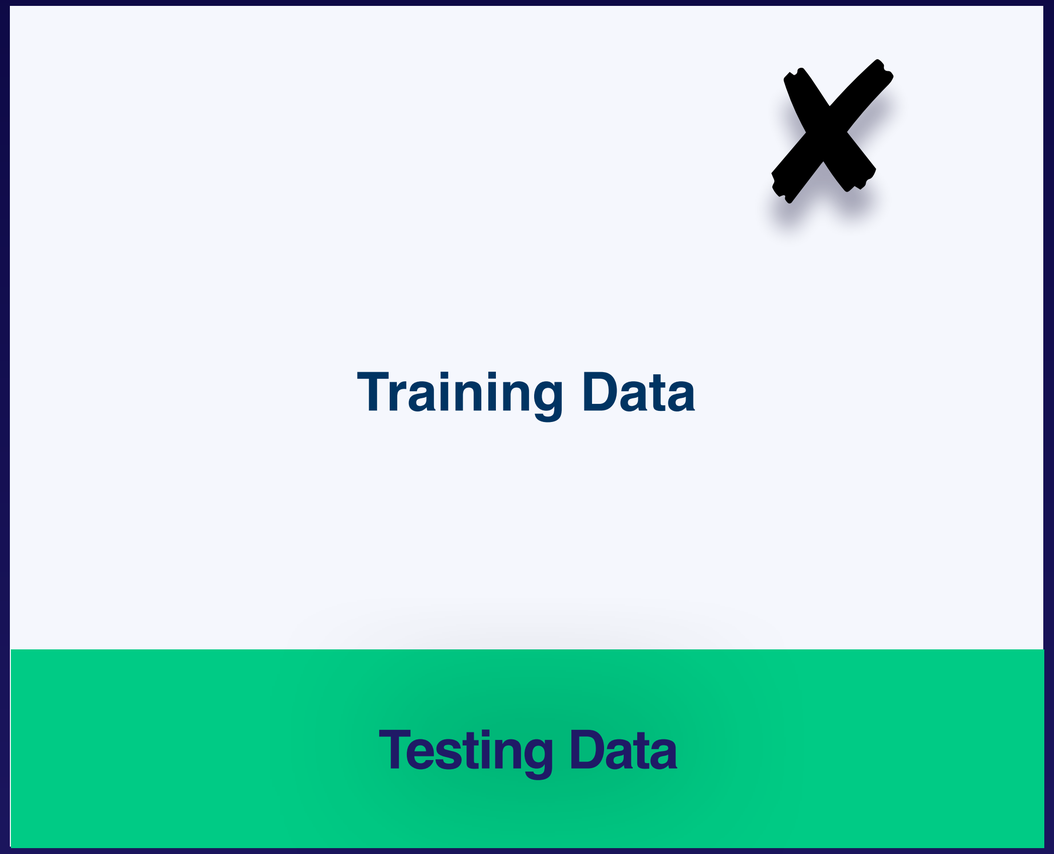


Data Preparation

{ 0,2,0,3,0,2,4…}

Our Data

{ 8,6,3,6,1,3,9…}



Data Preparation

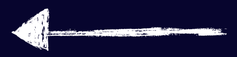
1. **Randomize**
2. Split
3. Train
4. Test

Data Preparation

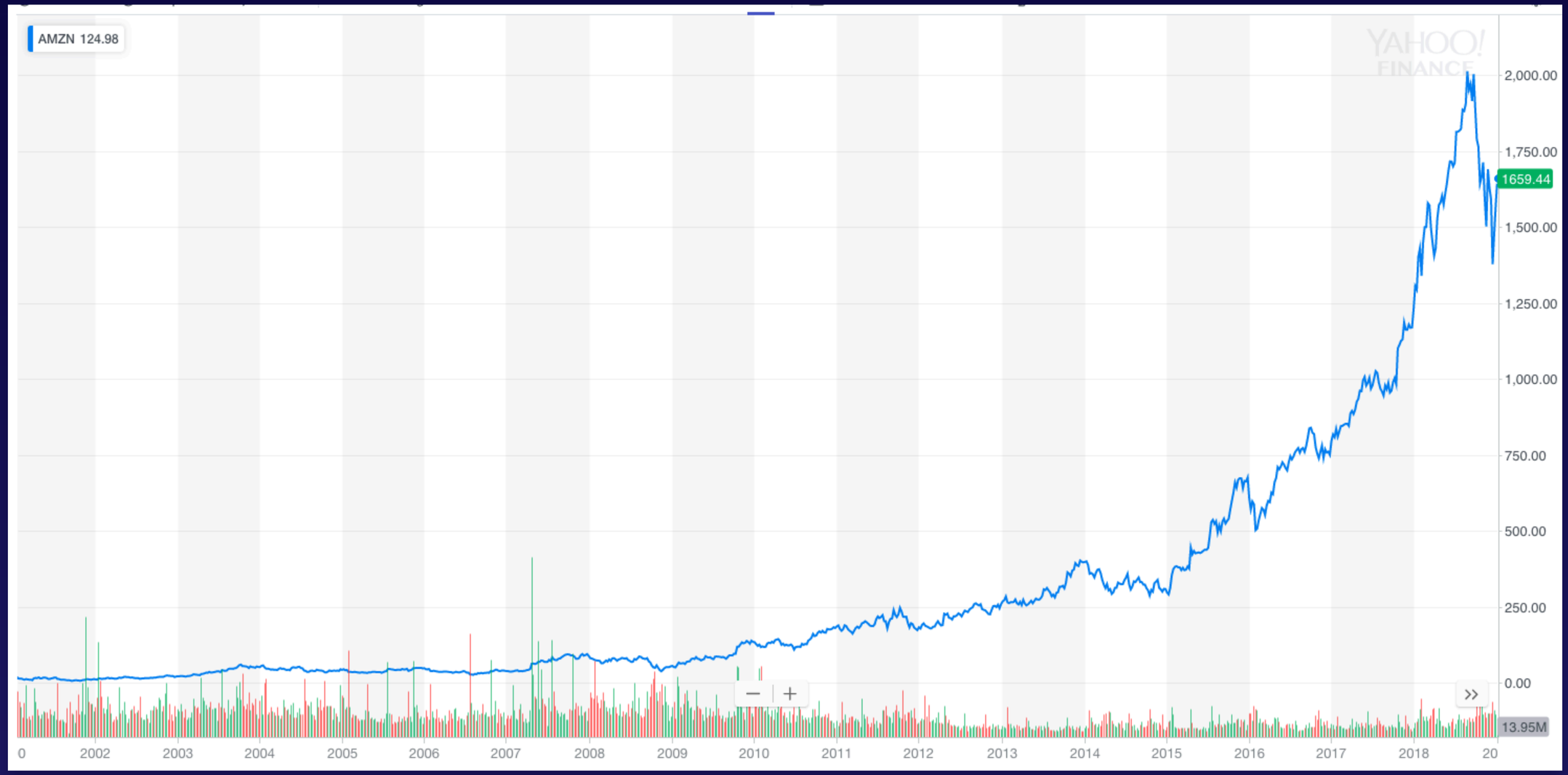
1. Randomize
2. Split
3. Train
4. Test

Data Preparation

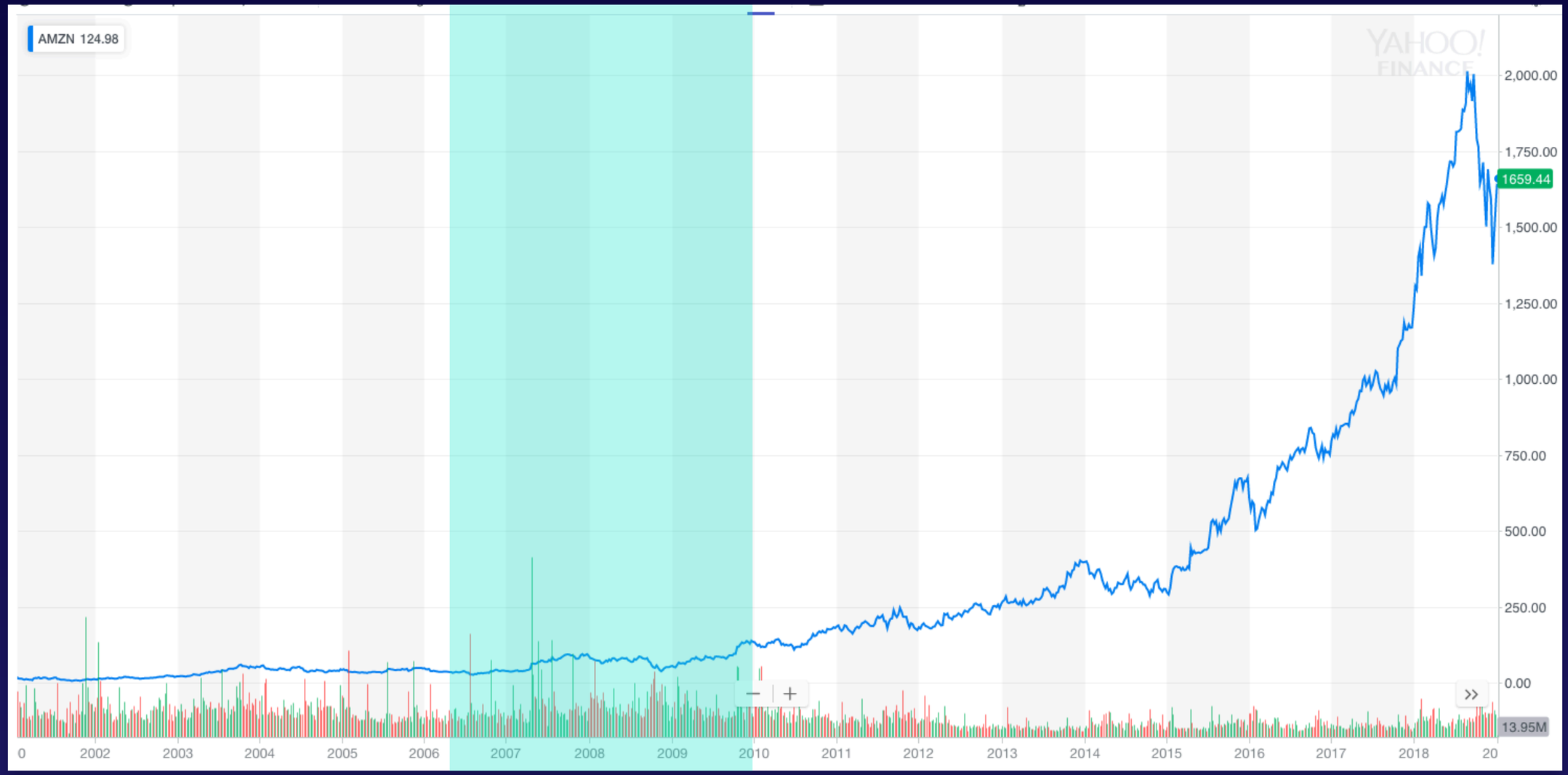

```
import numpy as np
import os


# read raw data
print("Reading raw data from {}".format(raw_data_file))
raw = np.loadtxt(raw_data_file, delimiter=',')

# split into train/test with a 90/10 split
np.random.seed(0)
np.random.shuffle(raw)
train_size = int(0.9 * raw.shape[0])
train_features = raw[:train_size, :-1]
train_labels = raw[:train_size, -1]
test_features = raw[train_size:, :-1]
test_labels = raw[train_size:, -1]
```



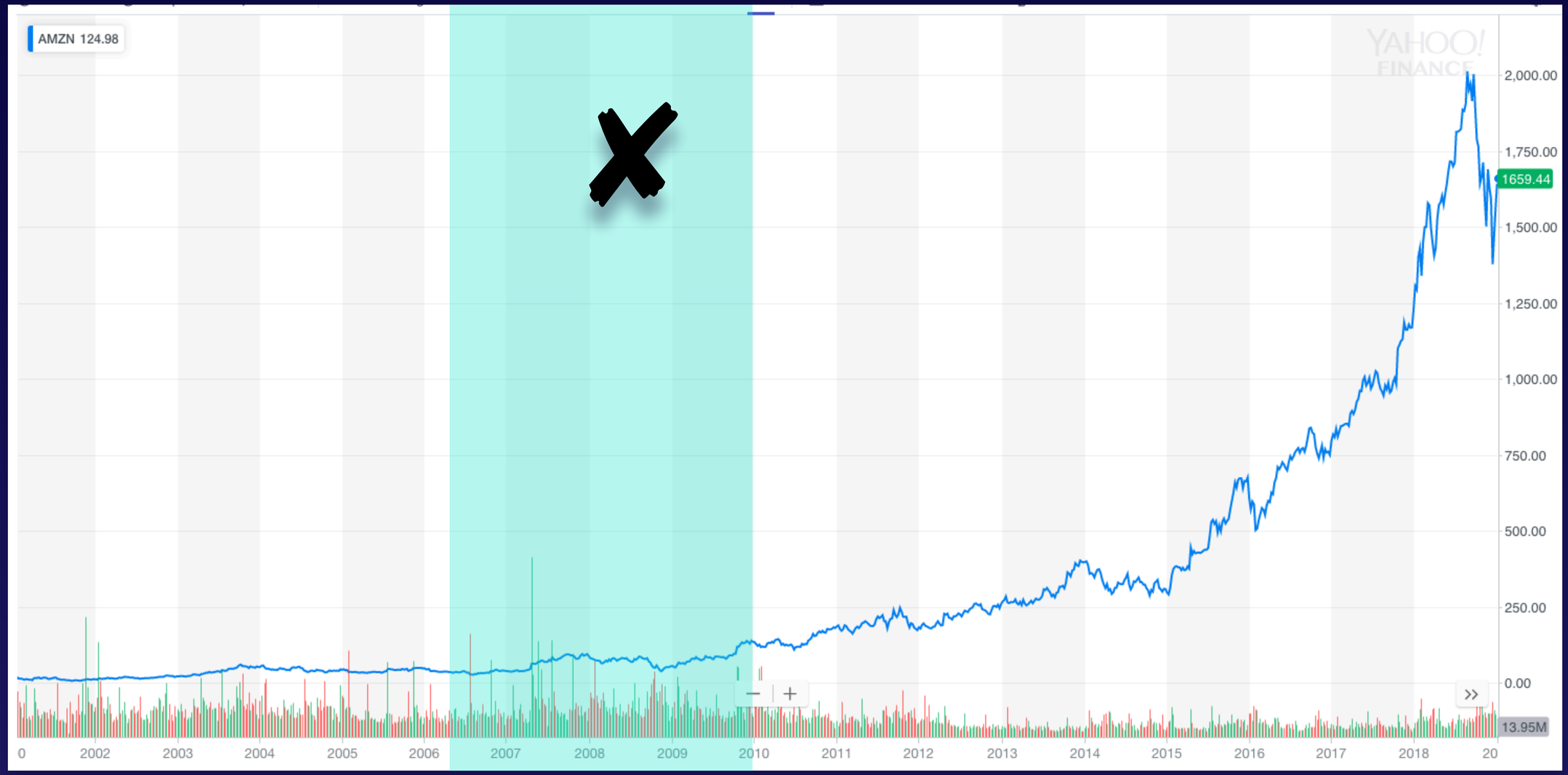
Data Preparation



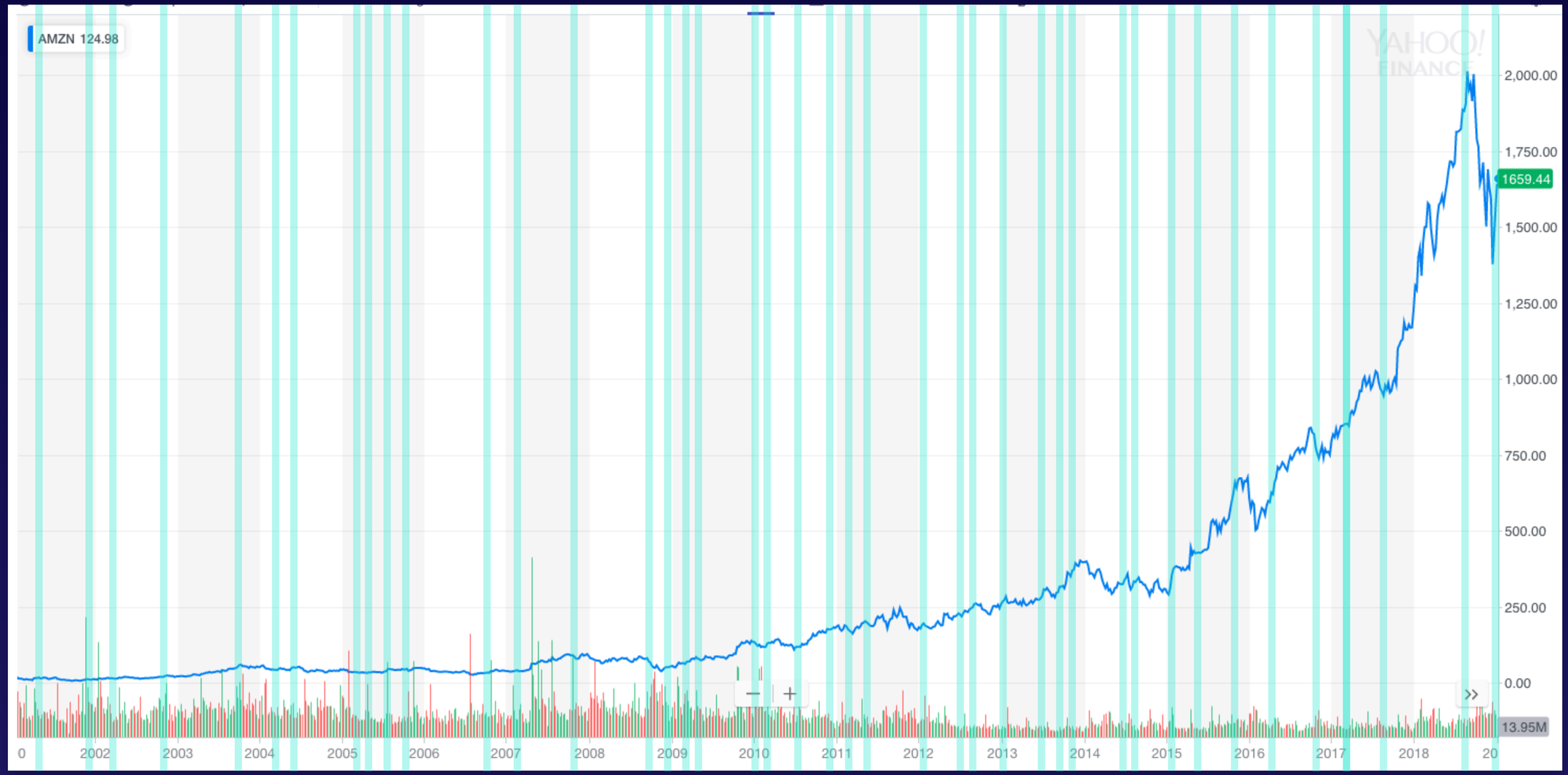
Data Preparation



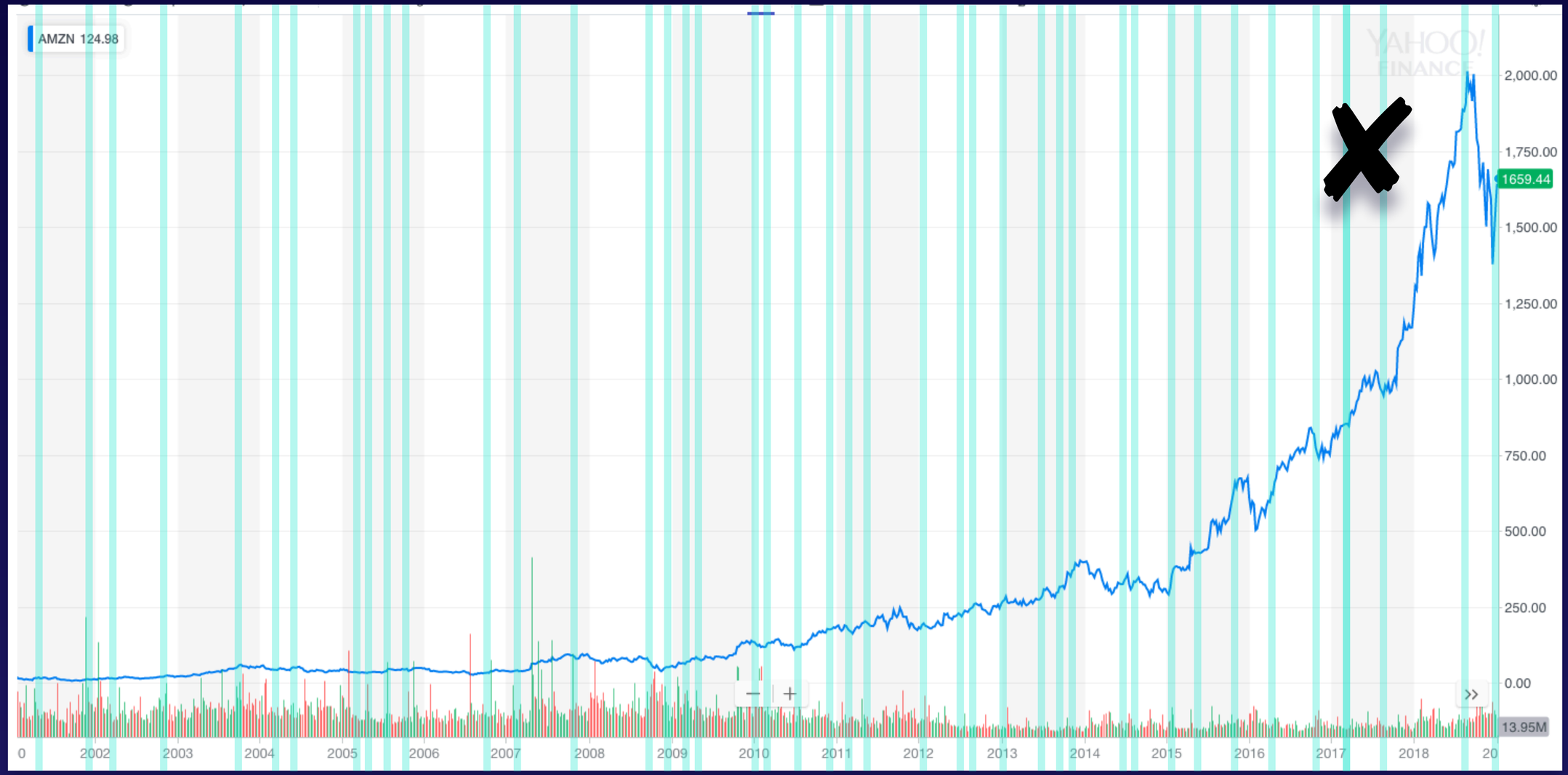
Data Preparation



Data Preparation



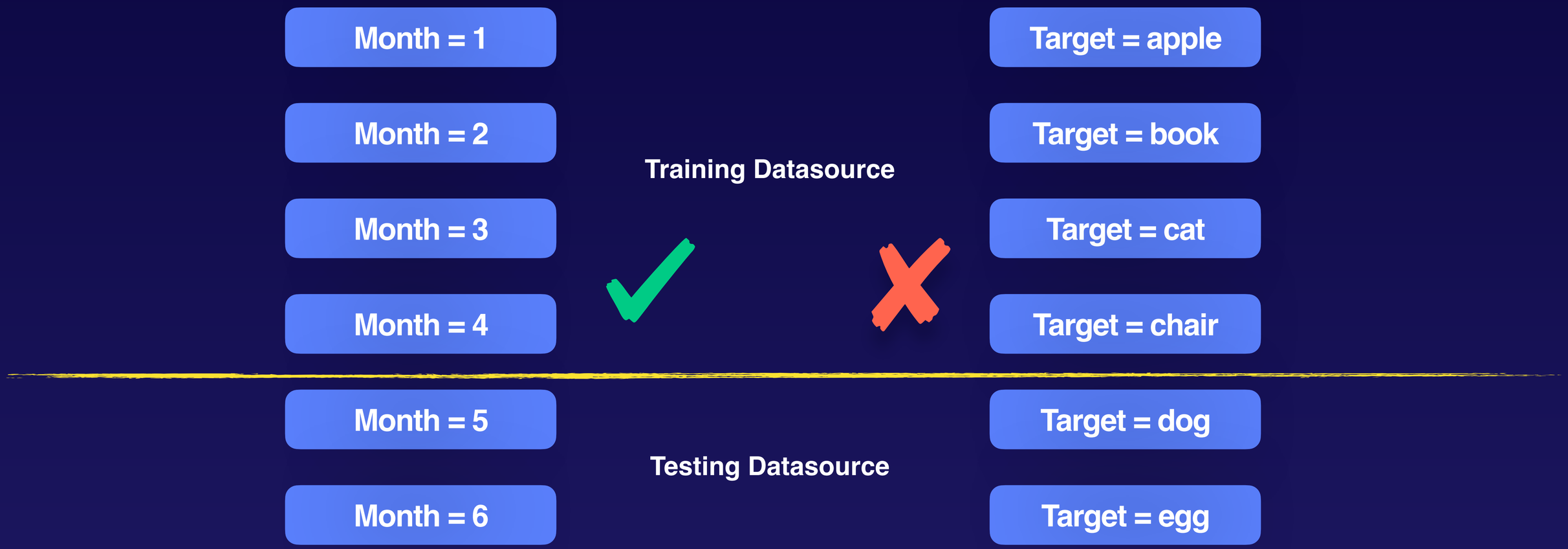
Data Preparation



Sequential Split Strategy

Month = 1		Target = apple
Month = 2	Training Datasource	Target = book
Month = 3		Target = cat
Month = 4		Target = chair
Month = 5	Testing Datasource	Target = dog
Month = 6		Target = egg

Sequential Split Strategy



Split Strategy

Movie: {.....}

Movie: {.....}

Movie: {.....}

Movie: {.....}

Movie: {.....}

Movie: {.....}

Consider if you will...

A data set of movie ratings that looks perfectly random.

Let's use a sequential split.

Split Strategy

Movie: {.....}

Movie: {.....}

Movie: {.....}

Movie: {.....}

Training Datasource

Movie: {.....}

Movie: {.....}

Testing Datasource

Split Strategy

{ ….Genre : ‘Adventure’…. }

{ ….Genre : ‘Adventure’…. }

{ ….Genre : ‘Comedy’…. }

{ ….Genre : ‘Documentary’…. }

{ ….Genre : ‘Romance’…. }

{ ….Genre : ‘Thriller’…. }

But, if we look closer, it seems the data was sorted by the 30th attribute…Genre.

Split Strategy

{ ….Genre : ‘Adventure’…. }

{ ….Genre : ‘Adventure’…. }

{ ….Genre : ‘Comedy’…. }

{ ….Genre : ‘Documentary’…. }

{ ….Genre : ‘Romance’…. }

{ ….Genre : ‘Thriller’…. }

But, if we look closer, it seems the data was sorted by the 30th attribute…Genre.

Romance and Thriller are under-represented as genres

Split Strategy

{ ….Genre : ‘Adventure’…. }

{ ….Genre : ‘Adventure’…. }

{ ….Genre : ‘Comedy’…. }

{ ….Genre : ‘Documentary’…. }

{ ….Genre : ‘Romance’…. }

{ ….Genre : ‘Thriller’…. }

Genre information is useless!

Romance and Thriller are under-represented as genres

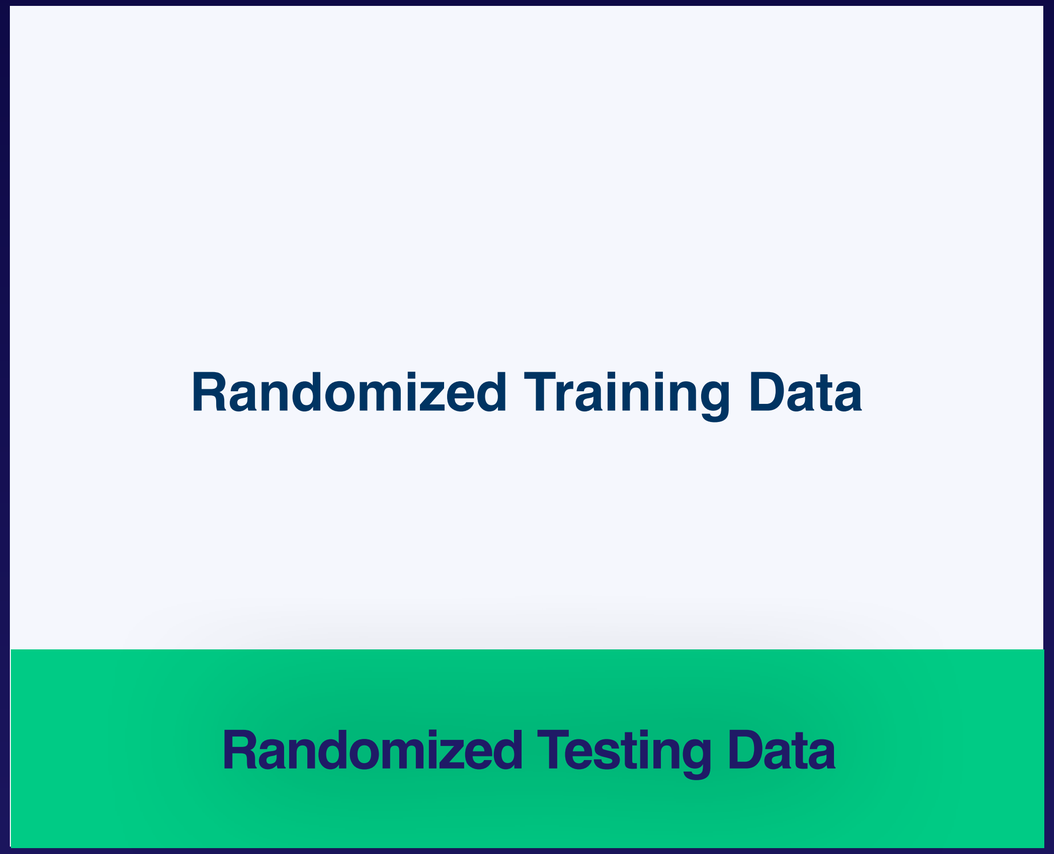
You Want to See Something REALLY Scary?

The model and evaluation set are too dissimilar by way of descriptive statistics to be useful.

This can happen when data is sorted by one of the columns in the dataset then split sequentially.

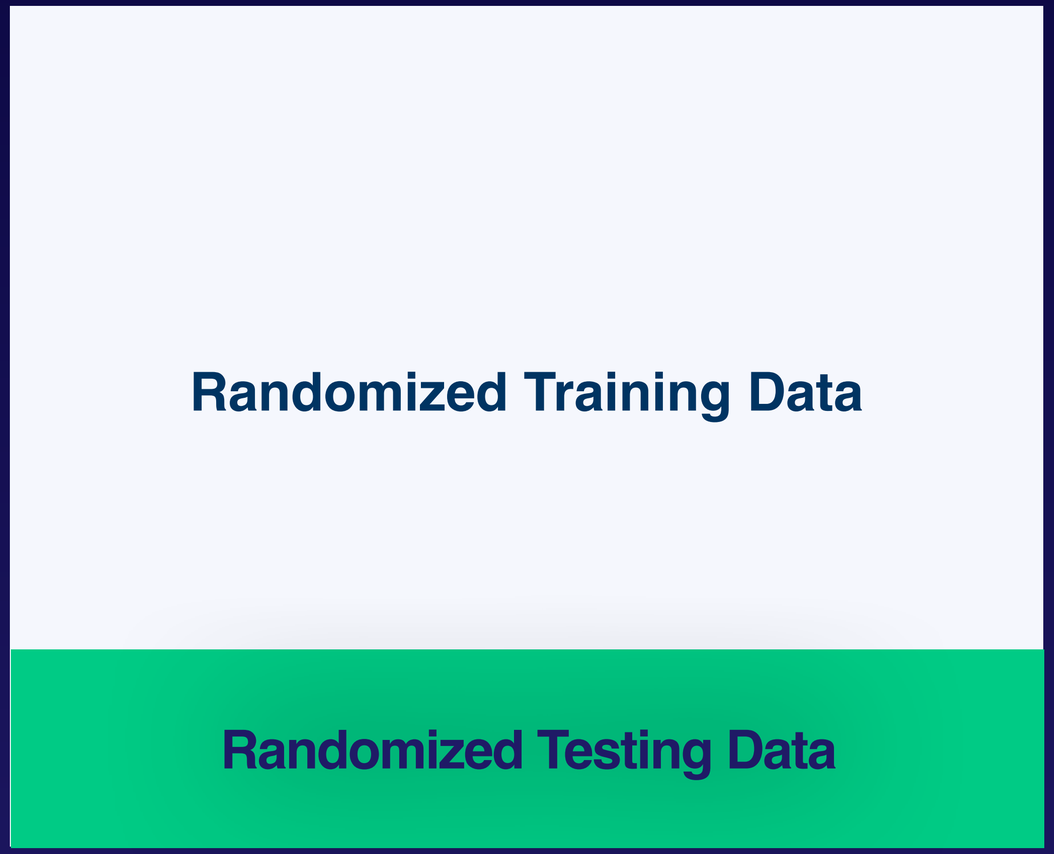
K-Fold Cross-Validation Method

1. Randomize
2. Split
3. Fold
4. Train
5. Test
6. Repeat



K-Fold Cross-Validation Method

1. Randomize
2. Split
3. Fold
4. Train
5. Test
6. Repeat



Round 1

K-Fold Cross-Validation Method

1. Randomize
2. Split
3. Fold
4. Train
5. Test
6. Repeat

Round 2

K-Fold Cross-Validation Method

1. Randomize
2. Split
3. Fold
4. Train
5. Test
6. Repeat

Round 3

K-Fold Cross-Validation Method

1. Randomize
2. Split
3. Fold
4. Train
5. Test
6. Repeat

Round 4

K-Fold Cross-Validation Method

Error Rate of Rounds

Round 1 ≈ Round 2 ≈ Round 3 ≈ Round 4

K-Fold Cross-Validation Method

Error Rate of Rounds

Round 1 ≈ Round 2 ≈ Round 3 ≈ Round 4

Round 1 < Round 2 > Round 3 > Round 4

K-Fold Cross-Validation Method

Error Rate of Rounds

Round 1 ≈ Round 2 ≈ Round 3 ≈ Round 4

Round 1 < Round 2 > Round 3 > Round 4