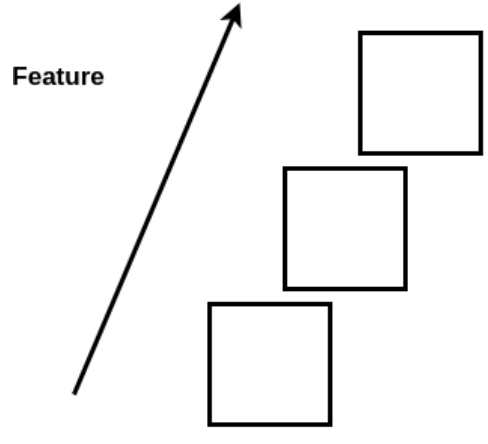
Shape of an example

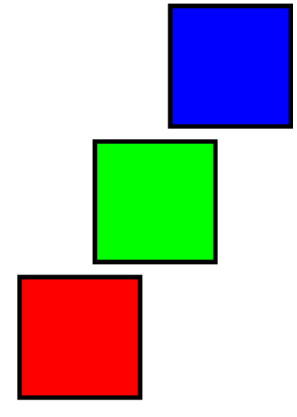
Up to now: each example is a *vector* of features

- this vector is *one-dimensional*: the only dimension is the feature dimension
- length 3: number of features

Zero non-feature dimensions, length 3 feature dimension



x



k

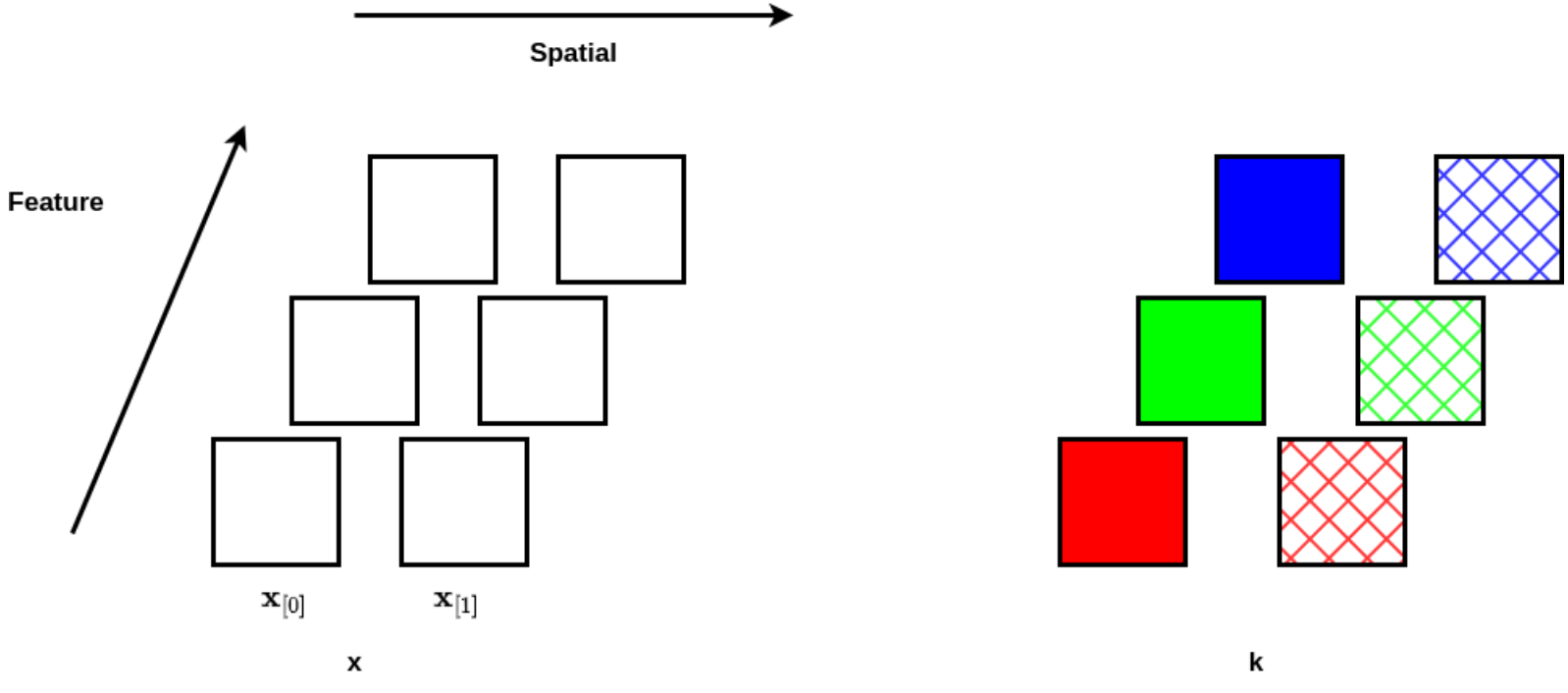
In subsequent lectures, we will learn about data that has *non-feature* dimensions

- time
- spatial

In the picture below, a *single example* $\mathbf{x}$

- has a single non-feature dimension ("location along a line")
- of length 2
- with vectors of features $\mathbf{x}_{[0]}$, $\mathbf{x}_{[1]}$ at each of the two locations
- the number of features is 3

One non-feature dimension of length 2, feature dimension of length 3



We could add a *second* non-feature dimension to represent *two-dimensional* space

- non-feature dimensions named "row" and "column"
- example
    - the pixel at spatial coordinates (R, C) have three features
        - intensity of the colors Red, Green, Blue

The diagram for *timeseries* would look similar to the one non-feature dimension example above

For example

- at each time step
- there are features: return, volume, momentum

There is an **critical distinction** between the feature dimension and non-feature dimensions

- there is **no ordering** in the feature dimension
- there **may be** ordering in the non-feature dimensions
    - temporal order: time occurs before time **+1**
    - element at spatial locations $(R, cC$ is to the northwest of element at location $(R+1, C+1)$

Moreover

- there is **no ordering** among examples
    - we can shuffle the order

What this means for "timeseries* data

- if each example represents a point in time
    - we have no way to capture this
- if we try to represent time in the feature dimension
    - we don't have the ability to actually impose order
    - the mathematics of the dot product are invariant to order
        - permuting the order of features and using an identical permutation for the ordering of parameters
        - yields the **same dot product**

In [2]:
```
print("Done")
```

```
Done
```