# Dimensionality Reduction Techniques

TM Quest

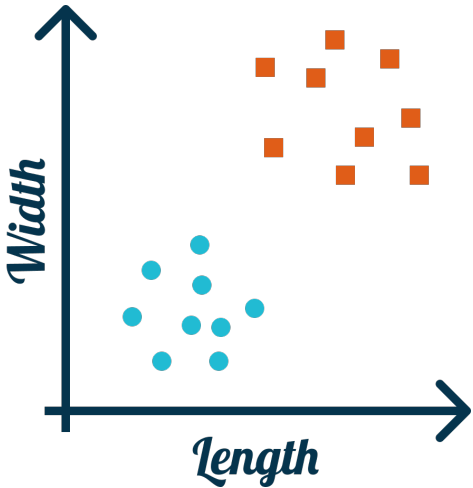
# Overview

## What Will we Learn in This Module?

- What is meant by reducing dimensions?
  - How to reduce dimensions based on correlation.
  - How to reduce dimensions based on variation.
- What are principal component analysis (PCA)?
  - For which problems are PCA useful?
  - How to implement PCA in scikit-learn.

# Dimensionality Reduction

# Idea




## Notice

- The more features you have, the higher the dimensionality of the problem is.
- We will henceforth refer to the number of features as the dimensionality of the problem we are trying to solve.

# The Curse of Dimensionality!

## Definition

Machine learning models tend to perform worse when the number of features used in the model increases. High-dimensional data is thus often difficult to predict well.

## Intuitive Explanations

- Imagine that you have three features that all can take the values 0 or 1. Then there are only $2 * 2 * 2 = 2^3 = 8$ different combinations. For every new observation, you have most likely seen several precisely equal data points in the training phase.
- Imagine now that you have 100 features that all can take the values 0 or 1. Then there are $2^{100} = 1,125,899,906,842,624$ different combinations. Given a new observation, you have probably not seen anything remotely close in the training phase.

# Solution: Reduce the Number of Dimensions

**Before**

| Feature 1 | Feature 2 | Feature 3 |
|---|---|---|
| 1 | 7 | 3 |
| 5 | 6 | 2 |
| 7 | 9 | 7 |
| 4 | 2 | 7 |
| 8 | 9 | 9 |
| 2 | 1 | 2 |
| 9 | 0 | 4 |
| 4 | 2 | 3 |

**After**

| Feature 1 | Feature 2 |
|---|---|
| 1 | 7 |
| 5 | 6 |
| 7 | 9 |
| 4 | 2 |
| 8 | 9 |
| 2 | 1 |
| 9 | 0 |
| 4 | 2 |

## Goal

Rather than removing features (columns) randomly, we would like to remove them based on some criterion.

# Principal Component Analysis

# Motivation

*Rather than simply removing some of the features, we can create a small set of new features that are combinations of the old ones.*

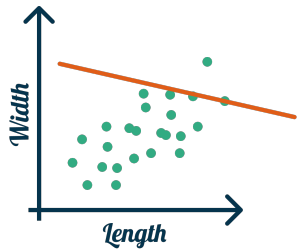
## Before

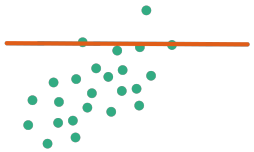
| Feature 1 | Feature 2 | Feature 3 |
|---|---|---|
| 1 | 7 | 3 |
| 5 | 6 | 2 |
| 7 | 9 | 7 |
| 4 | 2 | 7 |
| 8 | 9 | 9 |
| 2 | 1 | 2 |
| 9 | 0 | 4 |
| 4 | 2 | 3 |

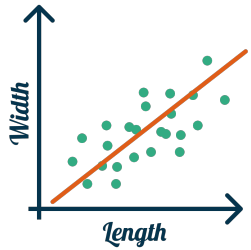
## After

| New Feature 1 | New Feature 2 |
|---|---|
| 8 | -2 |
| 11 | 3 |
| 16 | 0 |
| 6 | -3 |
| 17 | -1 |
| 3 | 0 |
| 9 | 5 |
| 6 | 1 |

# Principal Component Analysis






# Principal Component Analysis

## Definition

Principal Component Analysis (PCA) tries to pick a combination of columns that retain the variance in the data as much as possible.

## Facts

- In scikit-learn there is a pre-built class that performs PCA. You can use this in a scikit-learn pipeline to avoid the curse of dimensionality.
- There is a parameter that dictates how many new columns you want. The optimal value can be found with hyperparameter search.
- It's a good idea to scale the data before performing PCA.