PCA

Curse of Dimensionality

- Imagine you had to find a red square somewhre along a 100m line. It wouldn't take too long to find it.
- Now imagine finding a red square of the same size amongst a 100mx100m square... a little more difficult.
- Now imagine finding that square in a 100m3 cube. Considerably more difficult.
- Now imagine you are trying to find that red square in n cubes of 100m3 size.
- Assume n tends to infinity, or at least a very large number.

When the number of features or dimensions are too large, we face a lot of problems.

1. It's very easy to overfit the the training data, since we can have a lot of assumptions that describe the target label (in case of supervised learning). In other words we can easily express the target using the dimensions that we have.
2. We may need to increase the number of training data exponentially, to overcome the curse of dimensionality and that may not be feasible.
3. In ML learning algorithms that depends on the distance, like k nearest neighbors, everything can become far from each others and

Goal:

- Reduce Dimensions

Problem:

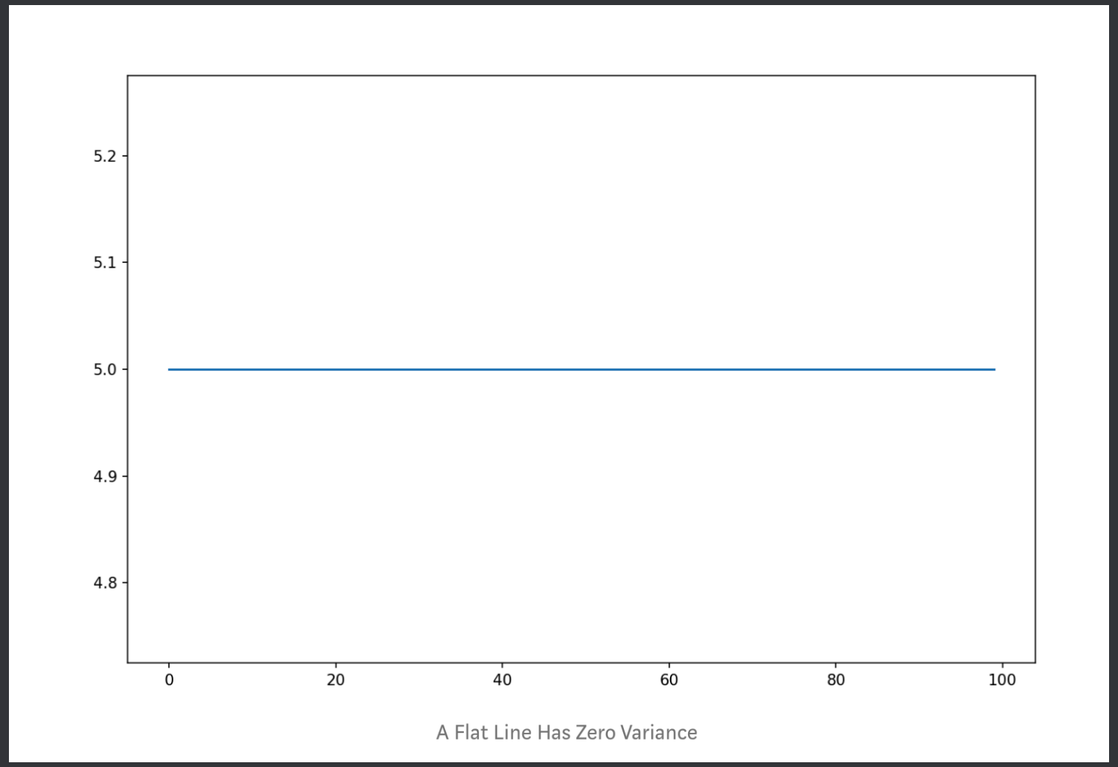
- We will lose information

Open Question:

- How to reduce dimensions without losing information

Variance?

Zero variance

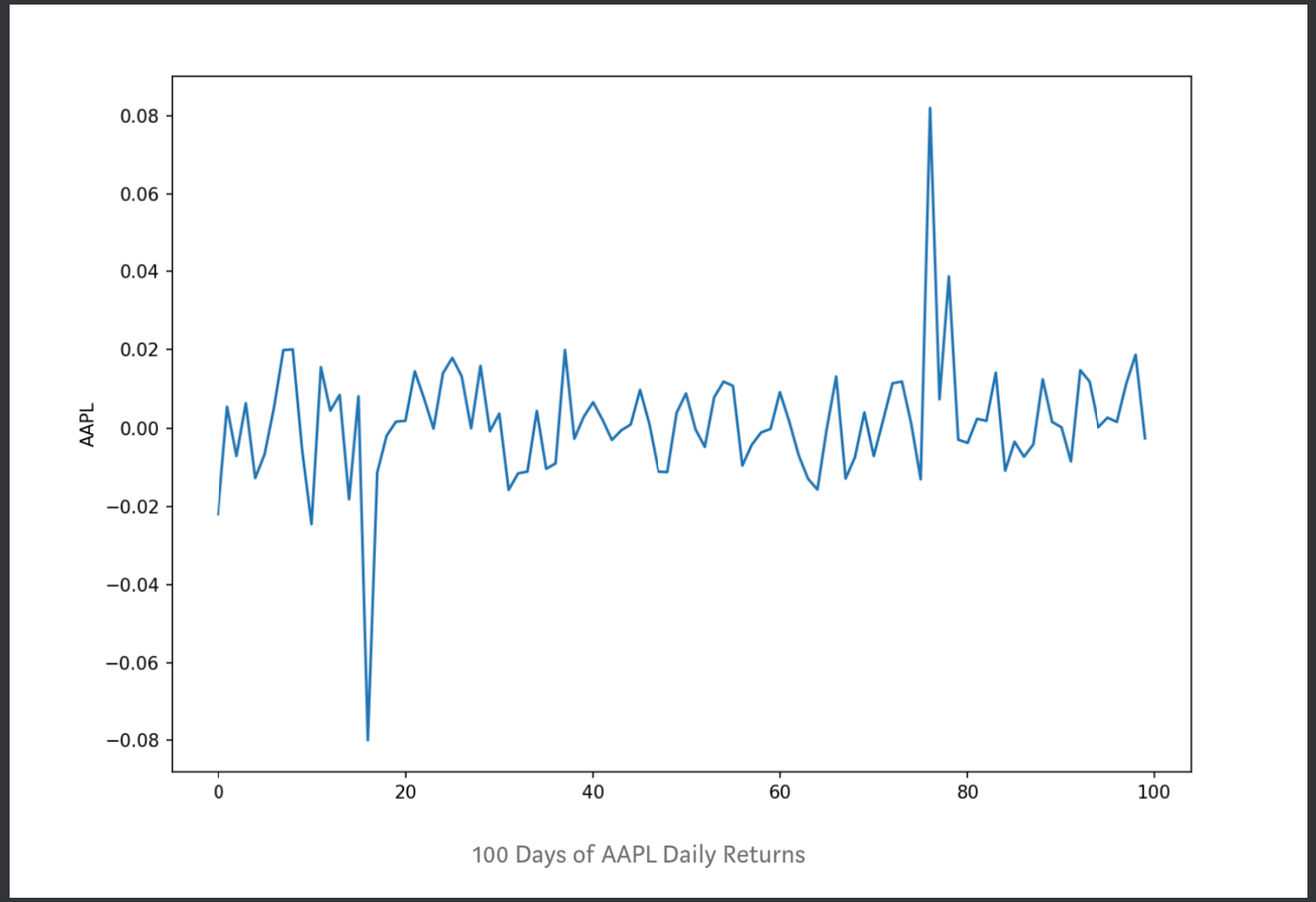


Zero variance

- Uninteresting
- Easy to predict
- Not much Value

Data with no variance has no uncertainty, so there is nothing for us to predict or explain

Variance


0.08
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
−0.08
AAPL
0
20
40
60
80
100
100 Days of AAPL Daily Returns

Variance

- Variance is basically how much the data bounces up and down by. The more bouncy it is, the harder it is to predict or explain.
- But amidst all that noise, bouncy data also contains signal (a.k.a. information).

Variance

Variance is friend and enemy

- Enemy because more variance in our target variable creates uncertainty and makes the target harder to predict.
- Friend though because, like we saw before, features with no variance are completely uninteresting and contain no signal at all. So to have a chance of building a good model, we need features with variance.

Capturing Signal with Principal Components

- We live in a world of too much information, not too little. The same holds true in data science — there is almost always a huge set of potential features we can use to make our prediction.
- We can’t use them all, due to high complexity.
- Without domain expertise, it can be hard to decide which features are worth to keep in our model.
- We wish there could a algorithm which could tell us that.

Ideal Set of Features

- **High Variance**: Features with lot of Variance contain lot of information.
- **Uncorrelated**: Highly correlated features are not useful, as they tell us same information
- **Not too Many**: Curse of Dimensionality

Solution

Principal Component Analysis

- PCA gives us our ideal set of features.
- It creates a set of principal components that are rank ordered by variance (the first component has higher variance than the second, the second has higher variance than the third, and so on) and orthogonal to each other meaning they are not correlated.
- Visualisation Tool