# Create and Analyze Features with Feature Engineering and Selection

**Guillermo Fernández**
DATA SCIENTIST

@guillermo_ai

# Summary

- Learn dimensionality reduction techniques for feature extraction
- Understand what Factor Analysis is
- Comprehend the most common clustering techniques
- Perform feature selection and feature engineering methods

# Extracting Features

# Principal Component Analysis (PCA)

**Converting and compressing data**

Into something that captures the essence of the original data

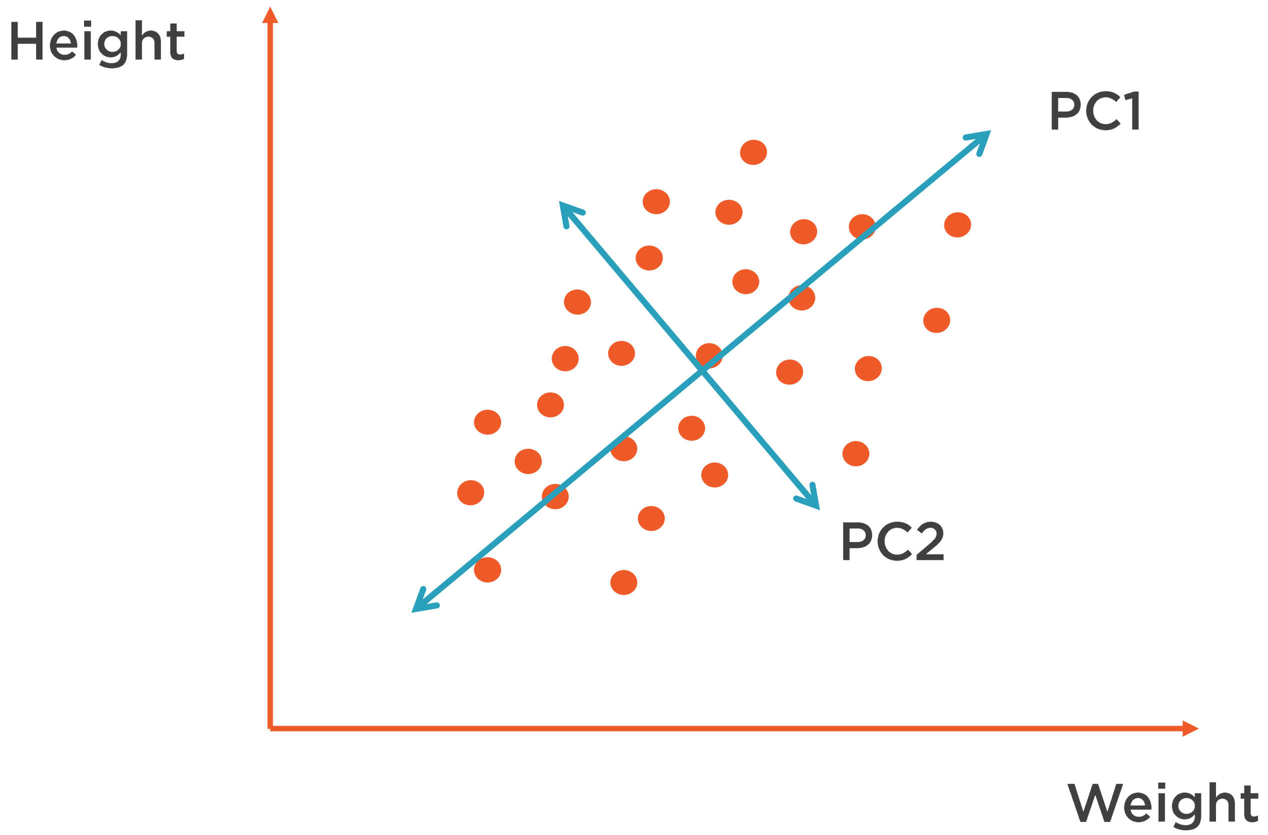
**Linear transformation algorithm**

Transformation into a new space

**Finds directions of maximum variance**

That are mutually orthogonal

# PCA Intuition



# Interpreting PCA

| Component | Eigenvalue | Proportion | Cumulative |
|---|---|---|---|
| 1 | 0.57 | 0.57/1.1 = 0.52 | 0.52 |
| 2 | 0.31 | 0.31/1.1 = 0.28 | 0.8 |
| 3 | 0.13 | 0.13/1.1 = 0.12 | 0.92 |
| 4 | 0.09 | 0.09/1.1 = 0.08 | 1 |
| Total | 1.1 | | |

# PCA Considerations

Needs feature scaling or mean normalization in order to have comparable range of values

Only captures linear correlations (although there exist non-linear adaptations)

Explains the variance in data

Closely related to Factor Analysis but less domain specific

# Non Linear Methods

**t-SNE**

t-distributed stochastic neighbor embedding

**SOM**

Self Organized Maps

# Demo

**Learn how to perform a PCA with Python**

**Using package:**

- Scikit-learn

# Factor Analysis

# Factor Analysis

Is a method to model or search observed variables in terms of a smaller number of influential underlying unobservable factors or latent variables.

# Goals of Factor Analysis

**Extract maximum common variance**

From all variables of the dataset

**Help interpreting data**

Identifying influential features, highlighting relations among observations

# Factor Analysis Intuition

$$Y_i = \beta_{i0} + \beta_{i1}F_1 + \beta_{i2}F_2 + \beta_{i3}F_3 + e_i$$

**Observed Variables**

Rythm

Noise

Instruments

**Factor**

Music Quality

# Factor Analysis Assumptions

No outliers in dataset

Dataset size greater than number of factors

Variables should not present perfect multicollinearity

Does not require homoscedasticity between the variables

# Factor Analysis Types

## Exploratory

Assumes any observed variable is associated with any factor

## Confirmatory

Assumes each factor is associated with certain subset of observed variables

# Factor Analysis Steps

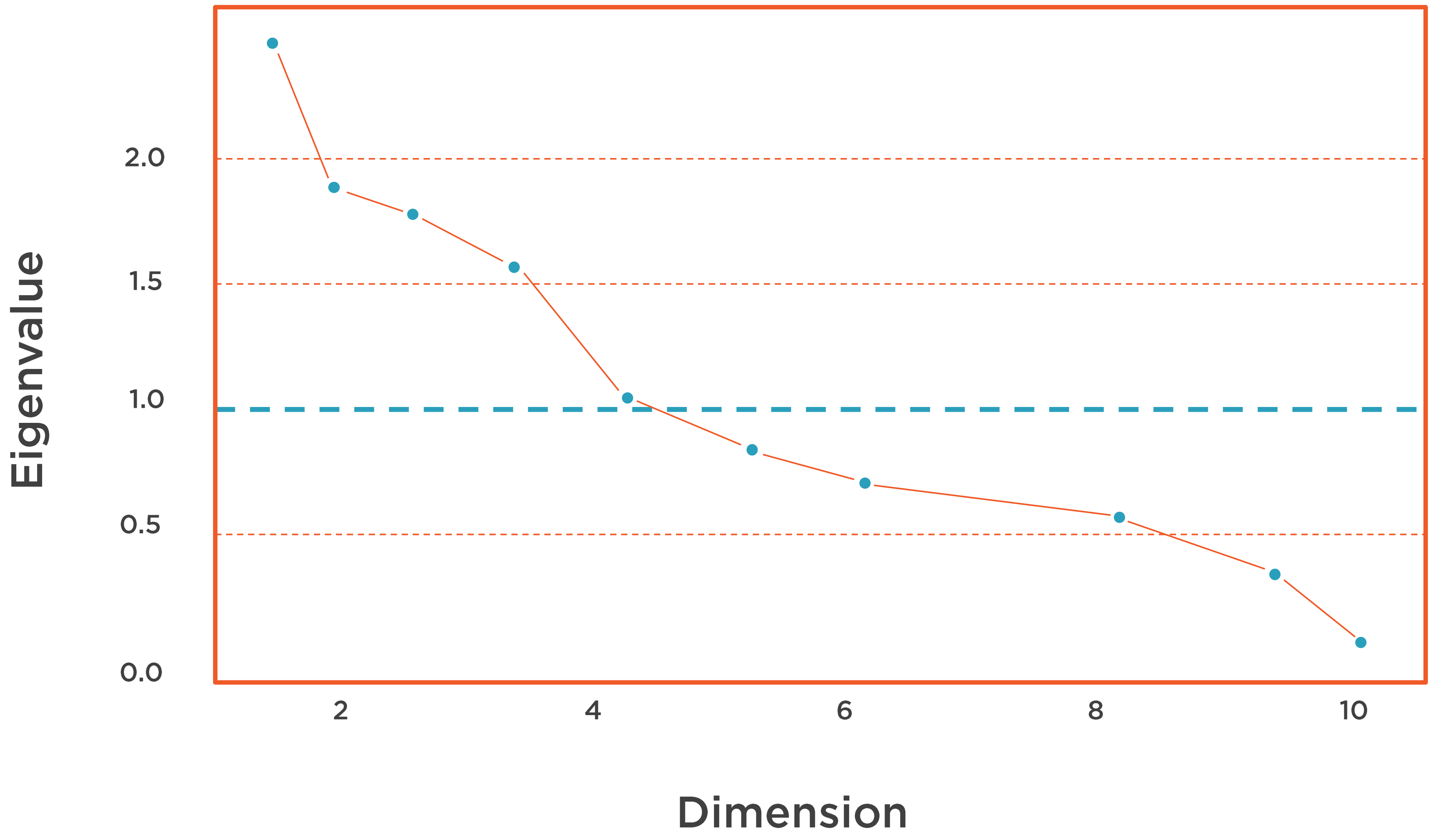
## Factor Extraction

Uses variance partitioning methods

## Factor Rotation

Tries to transform factors into uncorrelated factors for better interpretation

# Deciding the Number of Factors



# Comparison between PCA and FA

| PCA | FA |
|---|---|
| Explain maximum amount of variance | Explains covariance |
| Components are orthogonal | Orthogonality desired but not needed |
| Linear combination of observed variables | Linear combination of unobserved variables |
| Uninterpretable | Interpretable |
| Observational | Modeling technique |

# Demo

## Perform a Factor Analisis in Python

## Using package

- Scikit-learn
- Factor_Analyzer

# Clustering

# Clustering Goals

## Divide a dataset into natural groups

Previously undefined

## Describe unobserved groups

With the observed data

# Clustering Methods

**Hierarchical**

Agglomerative and Divisive

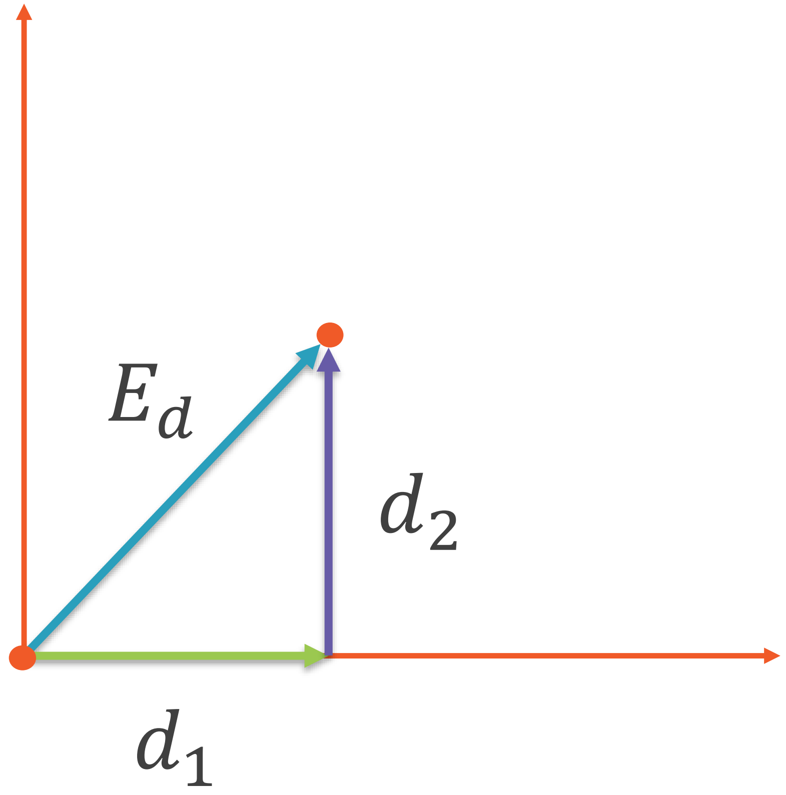
**Non Hierarchical**

K-means

**Model Based**

Uses a mixture model to specify the density function of variables

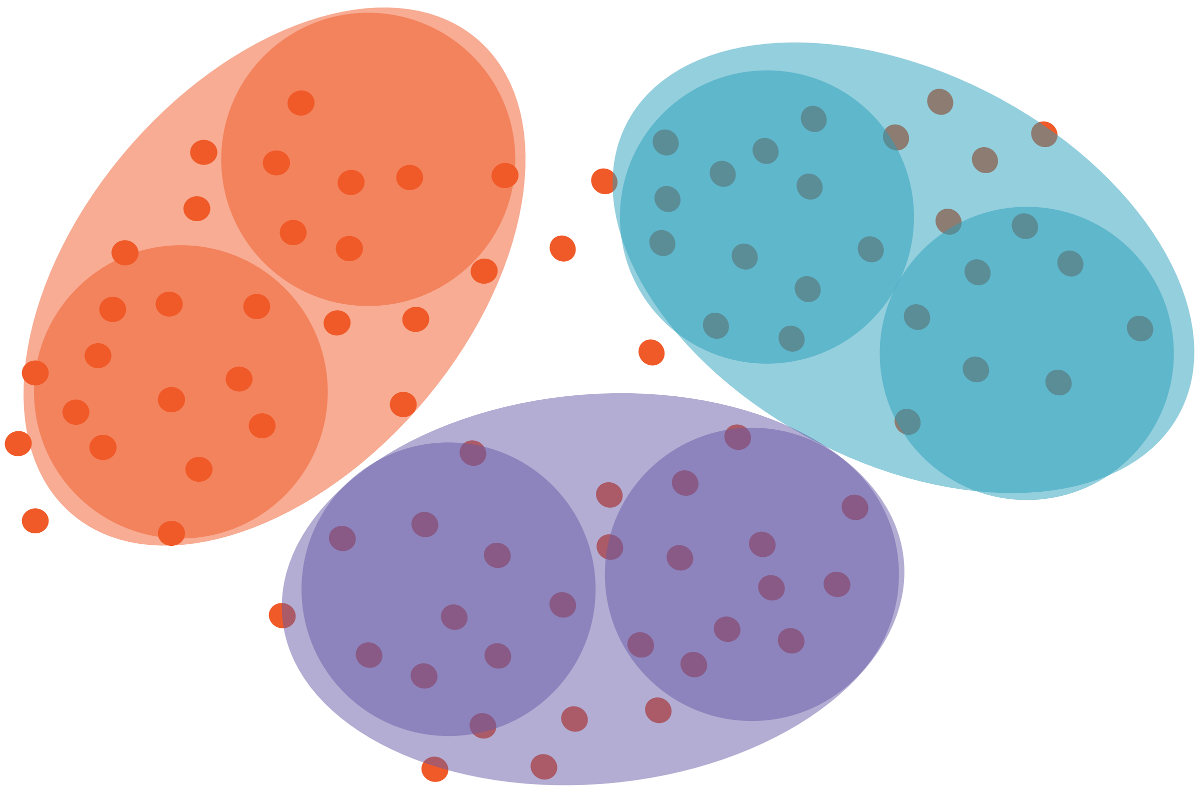
# Measures of Association



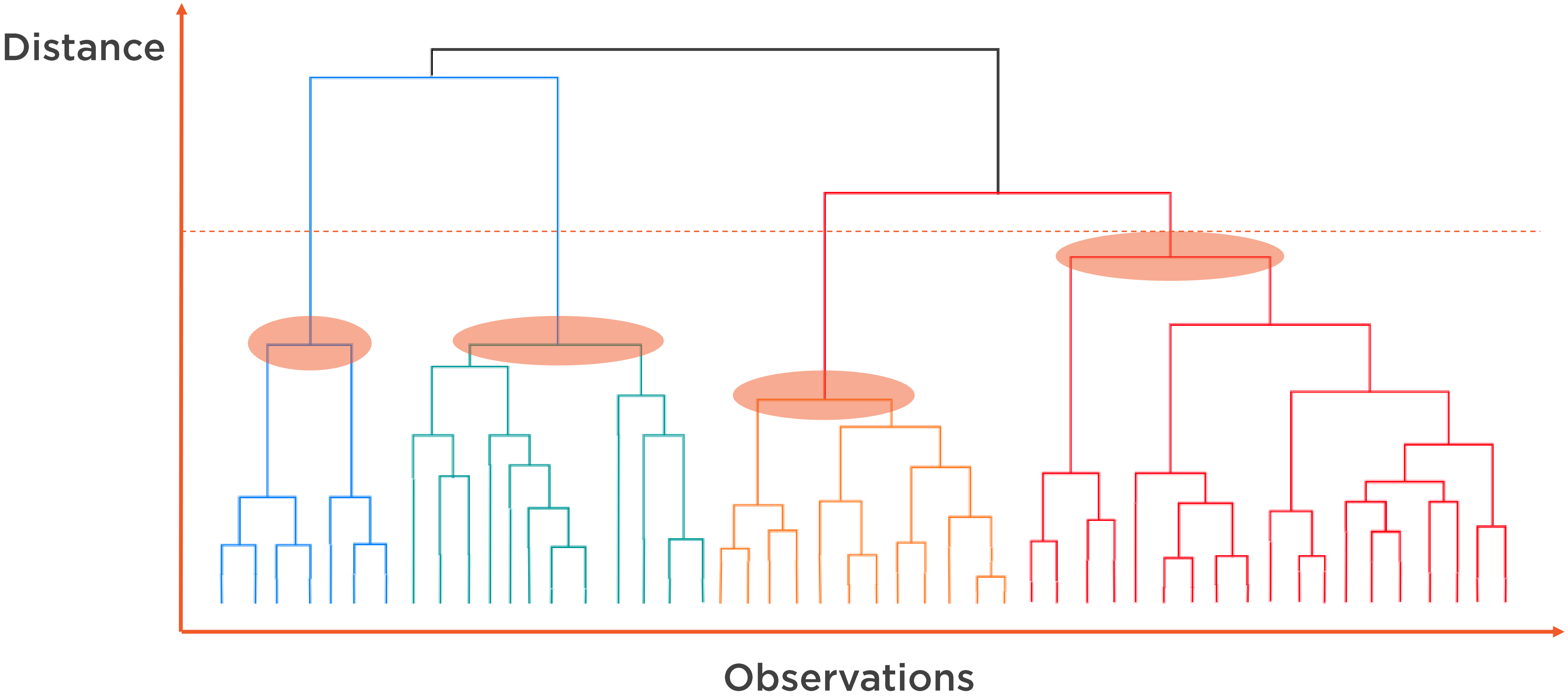

$$E_d = \sqrt{d_1^2 + d_2^2}$$

$$M_d = d_1 + d_2$$

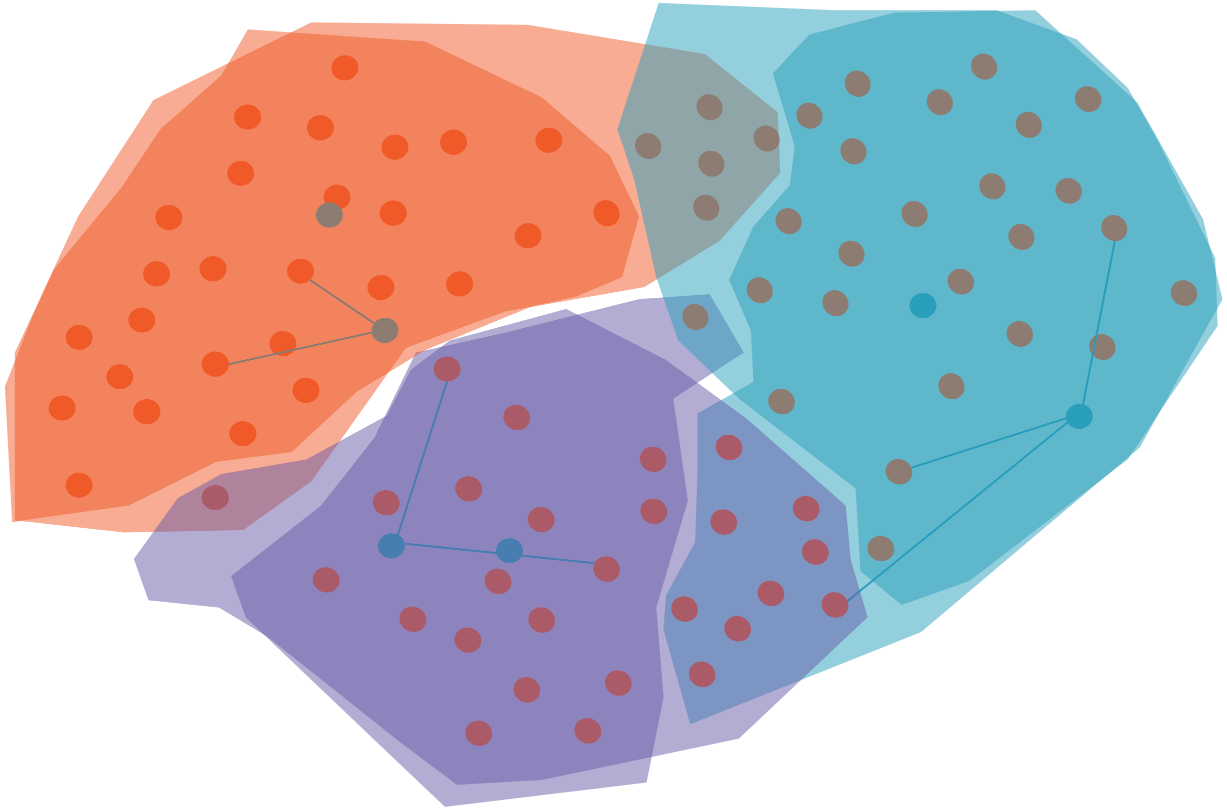
# Hierarchical Clustering



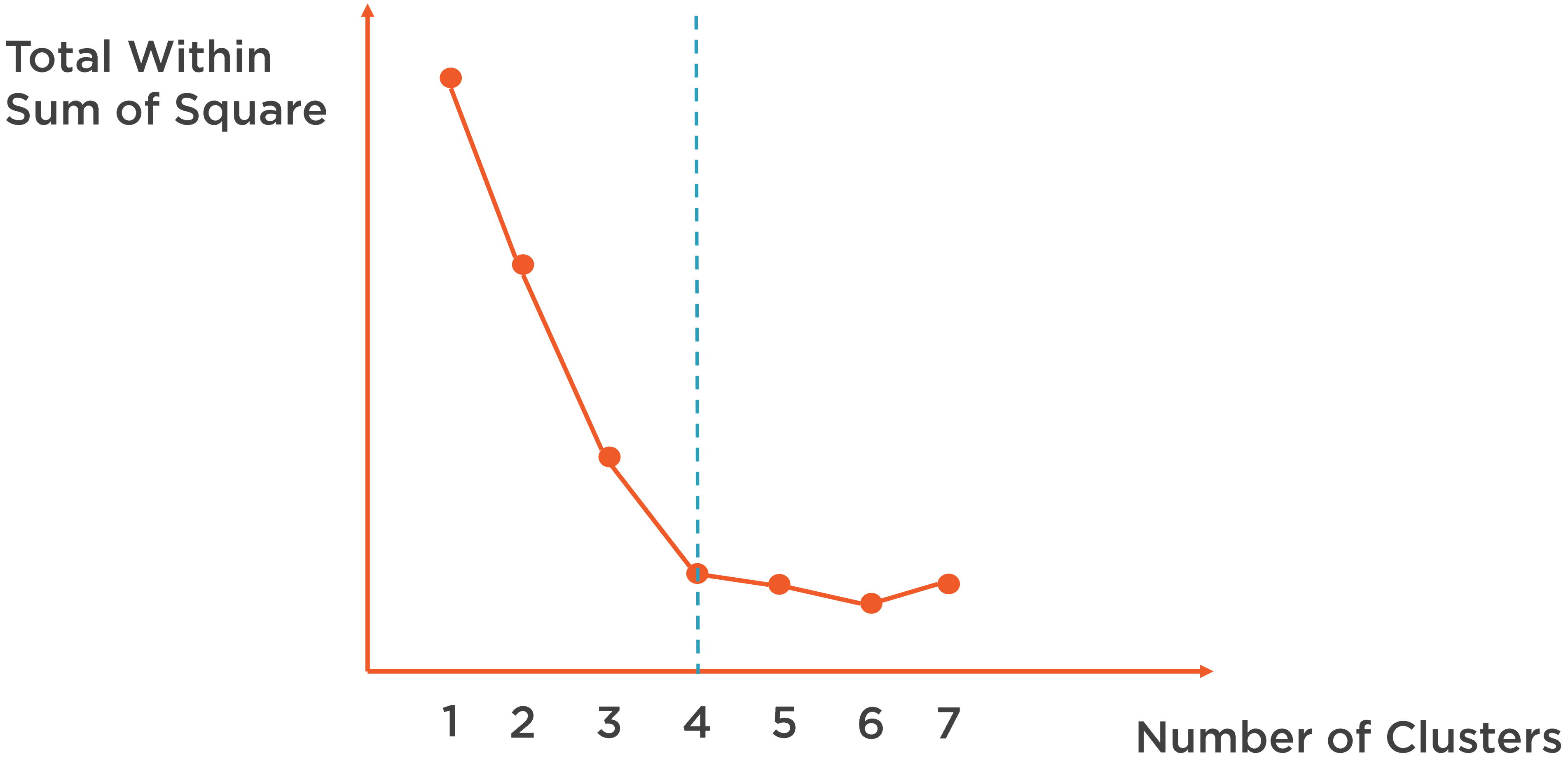
# Dendrogram – Tree Diagram



# K-Means



# Deciding the Number of Clusters



# Demo

**Perform K-Means and Hierarchical clustering techniques in Python**

**Using packages:**

- Scikit-learn
- Scipy

# Selecting Features

"More data beats clever algorithms, but better data beats more data."

**Peter Norvig**

# Goals of Selecting Features

**Identify**

Important features

**Remove**

Irrelevant and redundant features

**Improve**

Interpretability and predictive model performance

# Benefits of Selecting Features

Enables algorithms to train faster

Reduces complexity of a model

Improves accuracy of a model

Reduces overfitting

# Methods for Selecting Features

**Filter Methods**

Not based on models

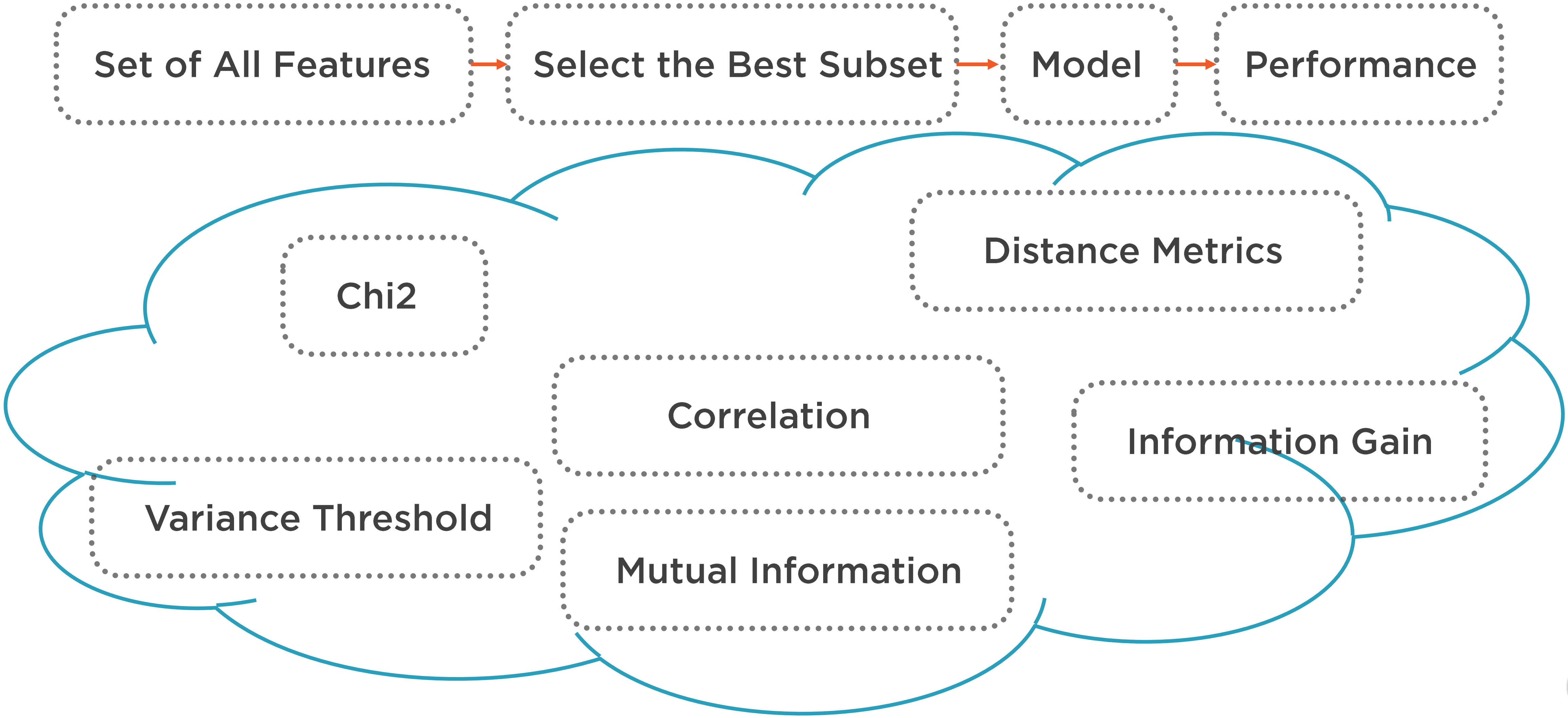
**Wrapper Methods**

Based on models

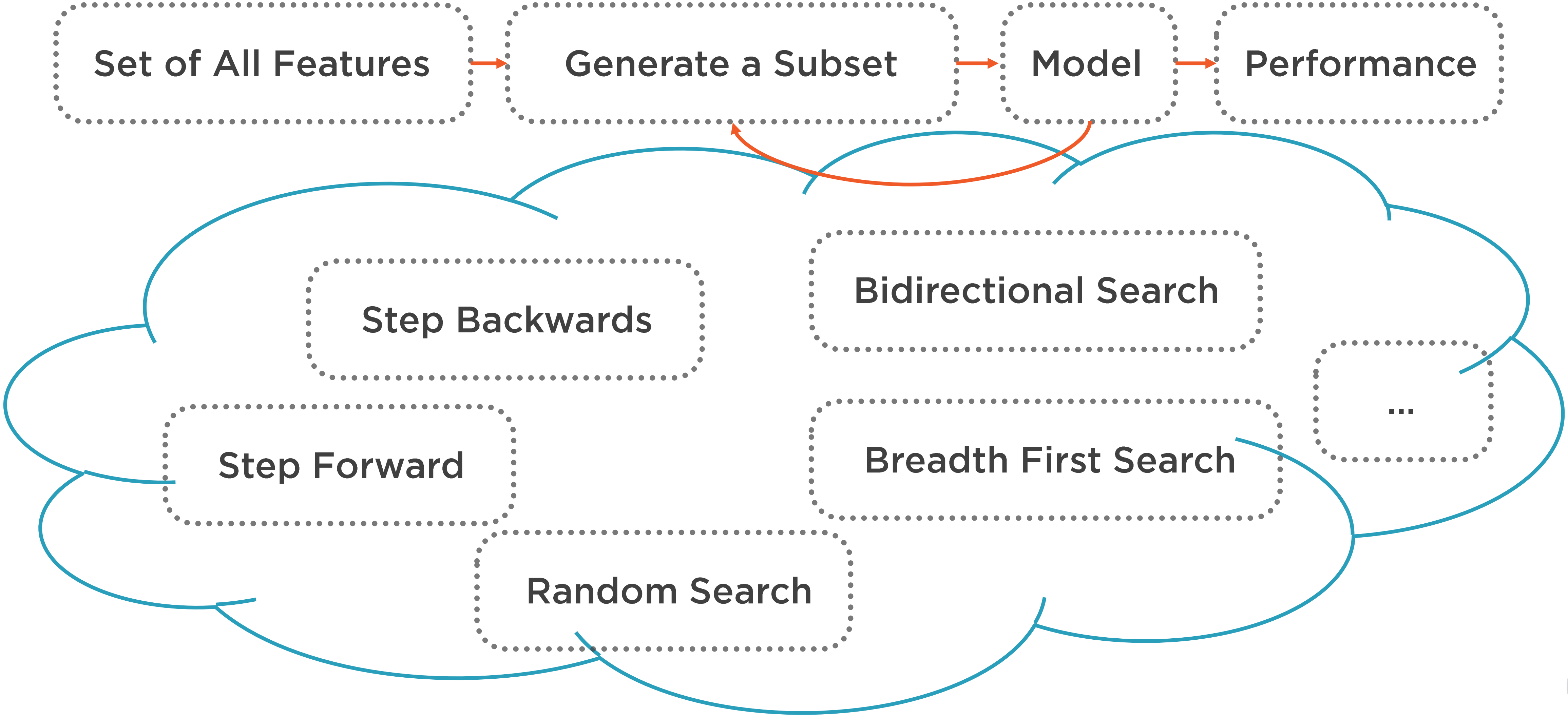
**Embedded Methods**

Based on models. Tries to combine filter and wrapper methods

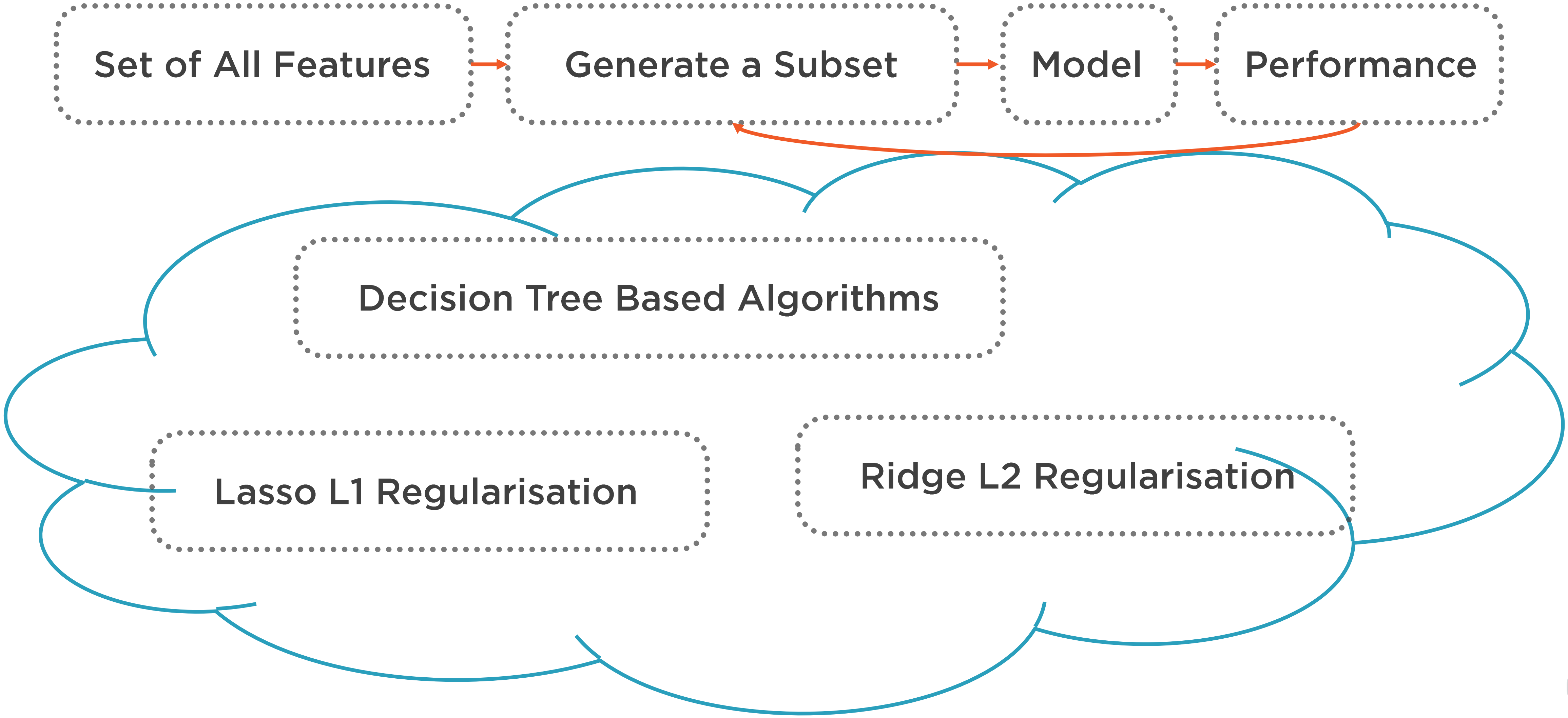
# Filter Methods



# Wrapper Methods



# Embedded Methods



# Demo

**Perform some of the most common Filter Methods for selecting features**

**Using packages:**

- Scikit-learn
- Scipy

# Engineering Features

"Is the process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data."

**Jason Brownlee**

"Coming up with features is difficult, time-consuming, requires expert knowledge.
Applied machine learning is basically feature engineering."

**Andrew Ng**

# Some Considerations

**Ideally at the begining**

But might have knowledge after performing EDA

**Is a representation problem**

How data is presented

**Feature engineering and selection**

Are not mutually exclusive

# Goals of Engineering Features

**Get the most out of your data**

For predictive modeling and data interpretation

**Improve and optimize**

Predictive model results

**Find the best representation of the data**

To learn a solution to a problem

# Benefits of Engineering Features

**Flexibility**

Less complex models, faster to run, easier to understand and mantain

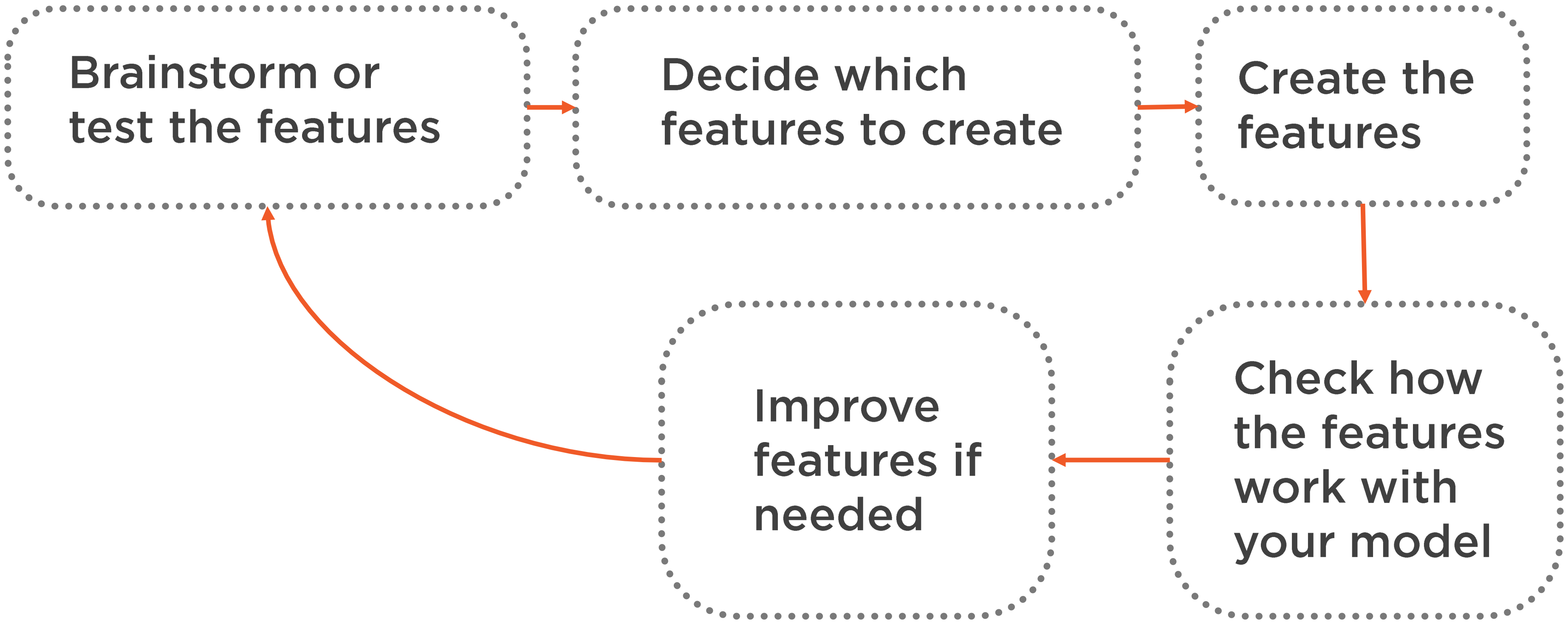
**Simpler models**

Easier to pick the most optimized parameters

**Better results**

Good features make you closer to the underlying problem

# Suggested Pipeline



# Feature Engineering Techniques

- Imputation
- Binning
- Log Transform
- Handling Outliers
- Feature Split
- One-Hot Encoding
- Feature Construction
- Grouping Operations
- Scaling
- Extracting Dates

Demo

**Perform some of the most common methods for engineering features**

**Using packages:**

- Pandas
- Numpy
- Scikit-learn
- Datetime