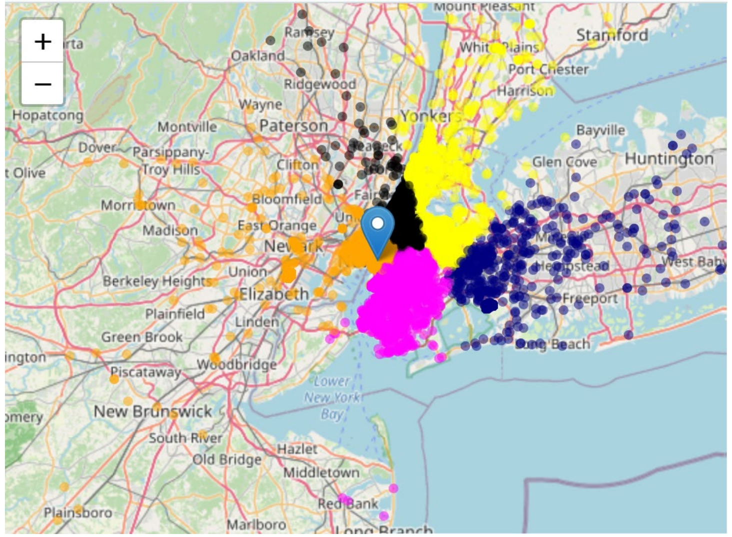
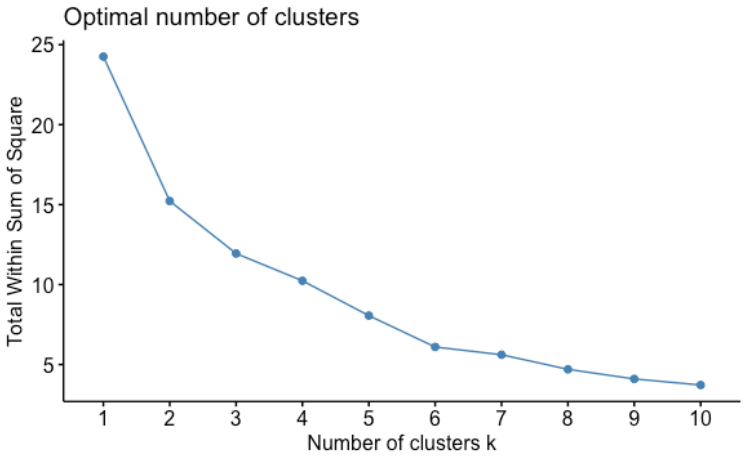
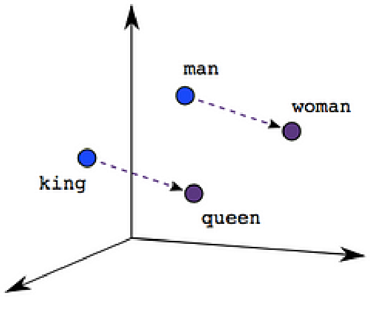
Unsupervised Learning

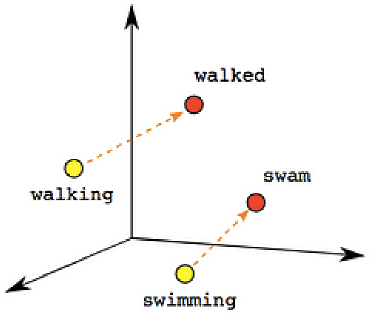
Business Intelligence



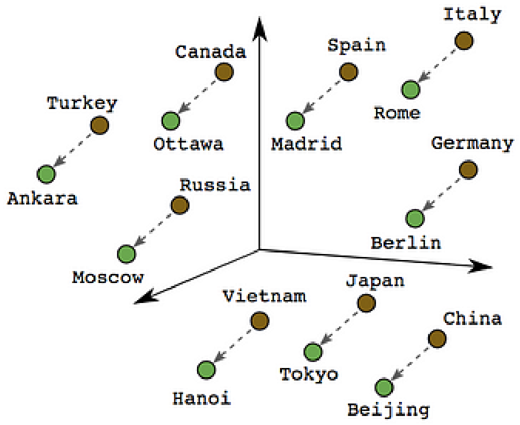




Male-Female



Verb Tense



Country-Capital

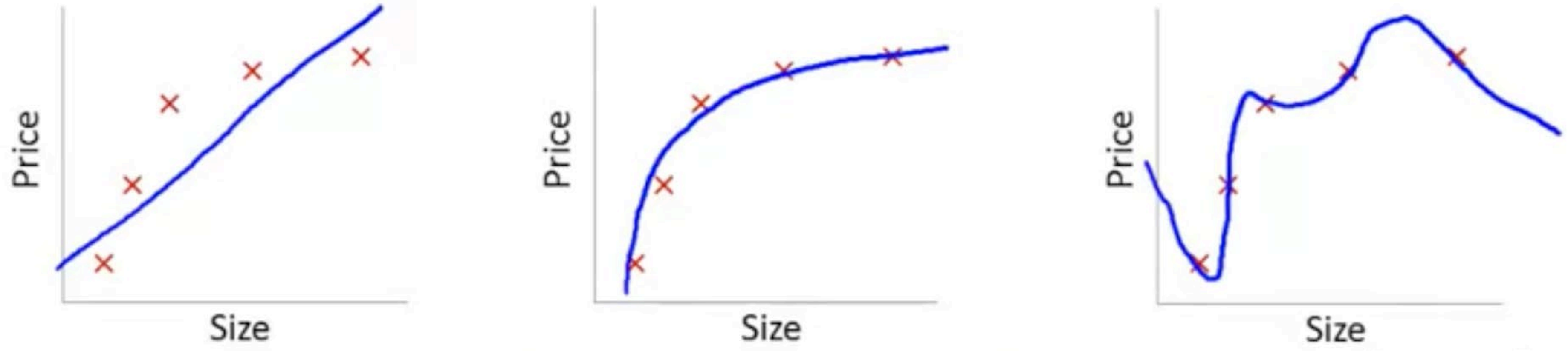
Terry College of Business
UNIVERSITY OF GEORGIA

One more Note on Supervised - Overfitting

Finding chance occurrences in the training dataset that look like interesting patterns, but that do not generalize to the population of interest, is called ***overfitting*** the data

Overfitting is therefore the tendency of tailoring models to the training dataset, ***at the expense of generalization to previously unseen data points***

Example: Linear regression (housing prices)

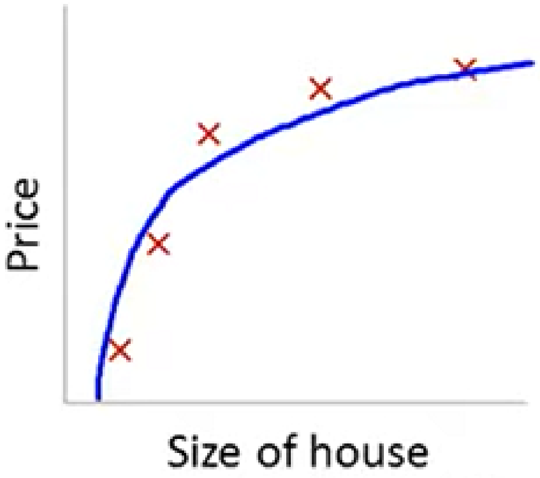


$\rightarrow \theta_0 + \theta_1 x$ "Underfit" "High bias"

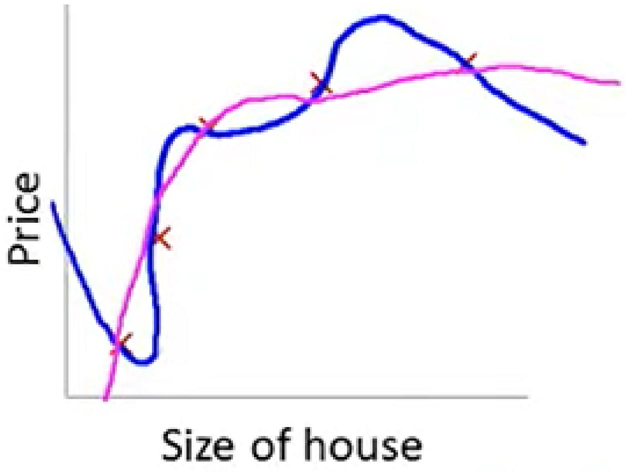
$\rightarrow \theta_0 + \theta_1 x + \theta_2 x^2$ "Just right"

$\rightarrow \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$ "Overfit" "High variance"

Regularization - Intuition

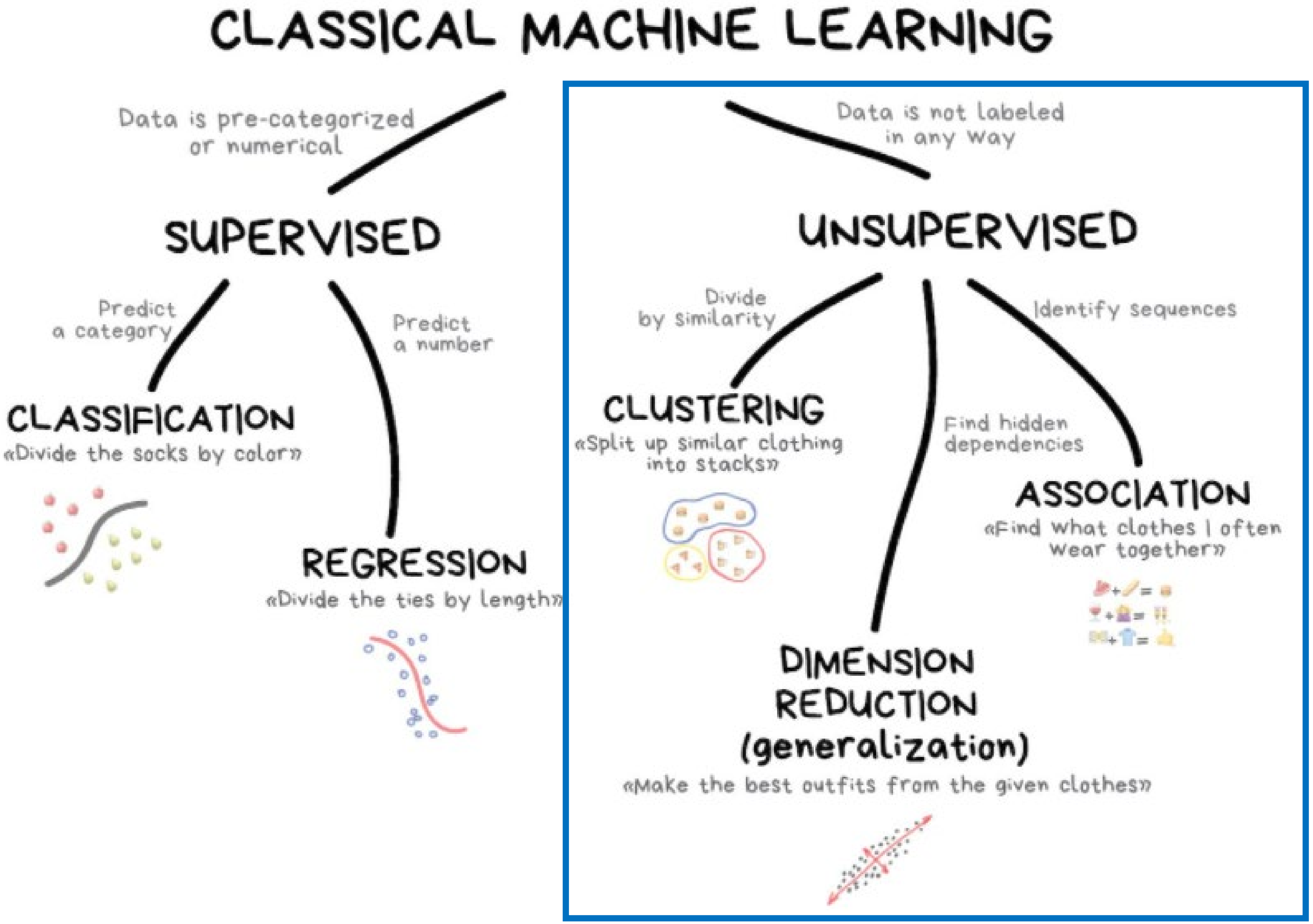


$$\theta_0 + \theta_1 x + \theta_2 x^2$$



$$\underline{\theta_0 + \theta_1 x + \theta_2 x^2} + \cancel{\theta_3 x^3} + \cancel{\theta_4 x^4}$$

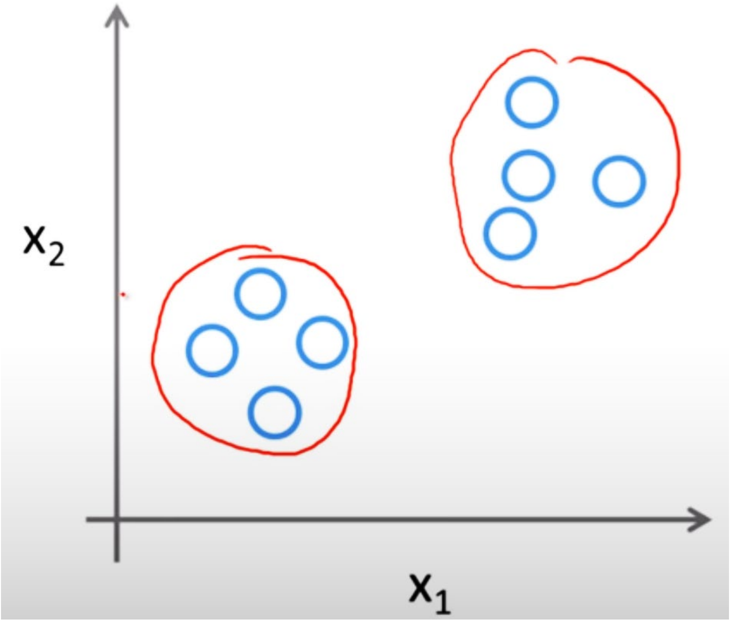
Suppose we penalize and make θ_3, θ_4 really small.



CLASSICAL MACHINE LEARNING
Data is pre-categorized or numerical
SUPERVISED
Predict a category
CLASSIFICATION
«Divide the socks by color»
Predict a number
REGRESSION
«Divide the ties by length»
Data is not labeled in any way
UNSUPERVISED
Divide by similarity
CLUSTERING
«Split up similar clothing into stacks»
Find hidden dependencies
Identify sequences
ASSOCIATION
«Find what clothes I often wear together»
DIMENSION REDUCTION (generalization)
«Make the best outfits from the given clothes»

Clustering

An example of *unsupervised learning* – main idea is to find groups of objects (e.g., customers), where the **objects within groups are similar** and **objects in different groups are not so similar**



Provost, F. & Fawcett, T. (2013). Data Science for Business.

Dimensionality Reduction

A method for representing data in a lower-dimensional space while still capturing the original data's essential information.

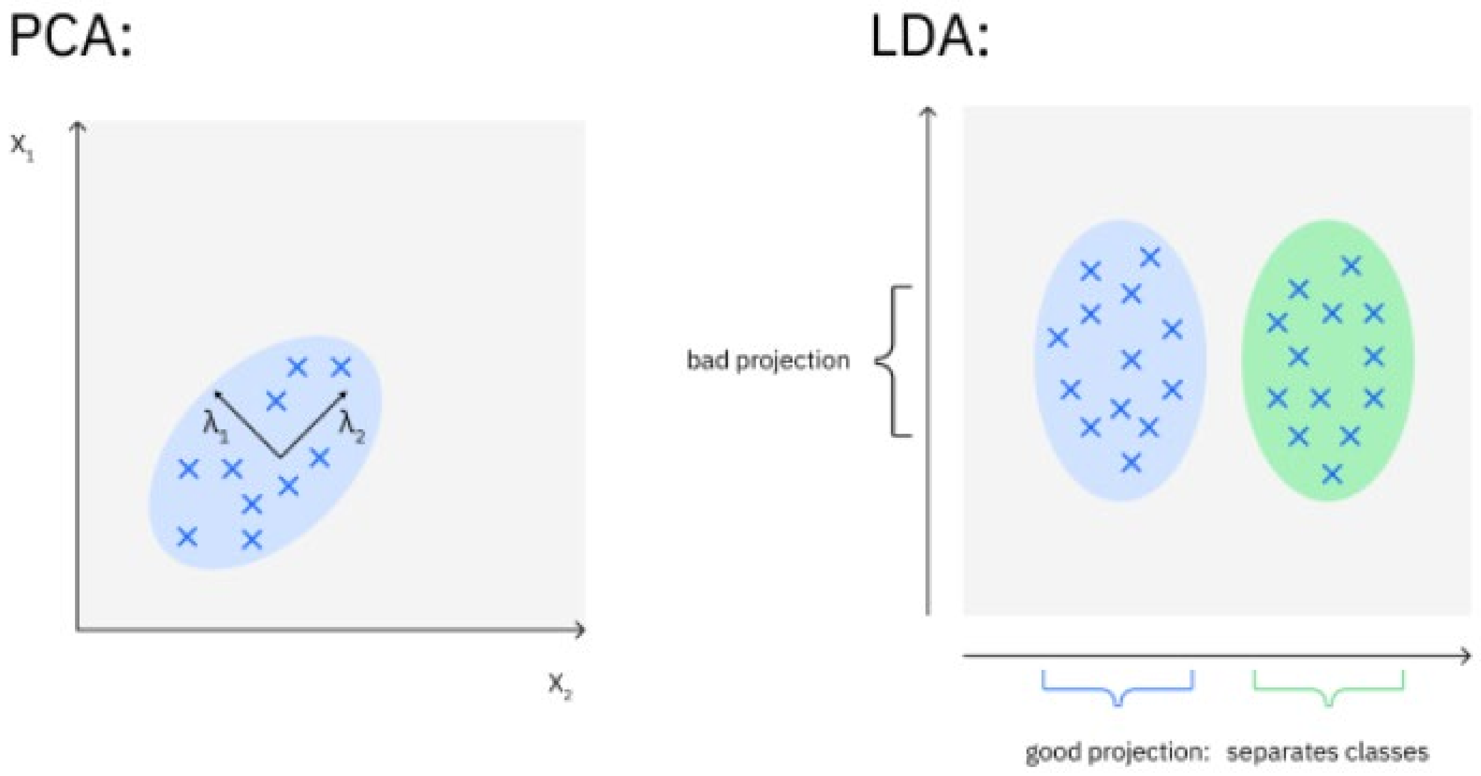
Why? **Simplify complex datasets**

- Data visualization – easier to visualize a dataset with 5 dimensions (or variables) than one with 50 dimensions.
- Decrease computation time and storage space for big data, etc.
- Decreased accuracy in predictive models trained on high-dimensional datasets - Curse of dimensionality.

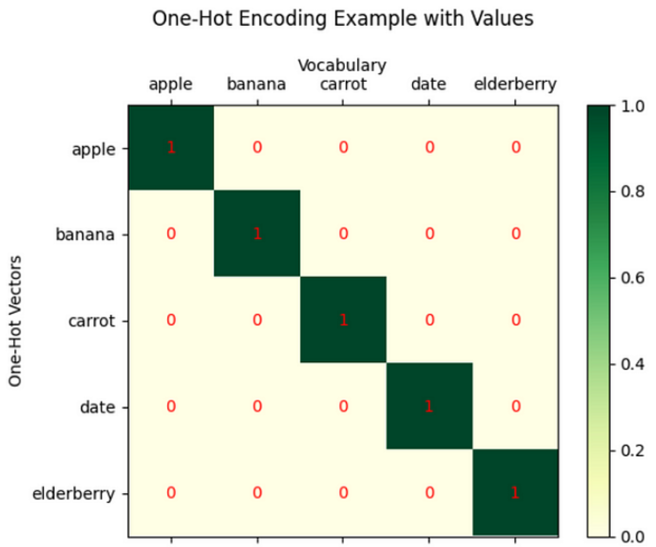
https://www.ibm.com/topics/dimensionality-reduction

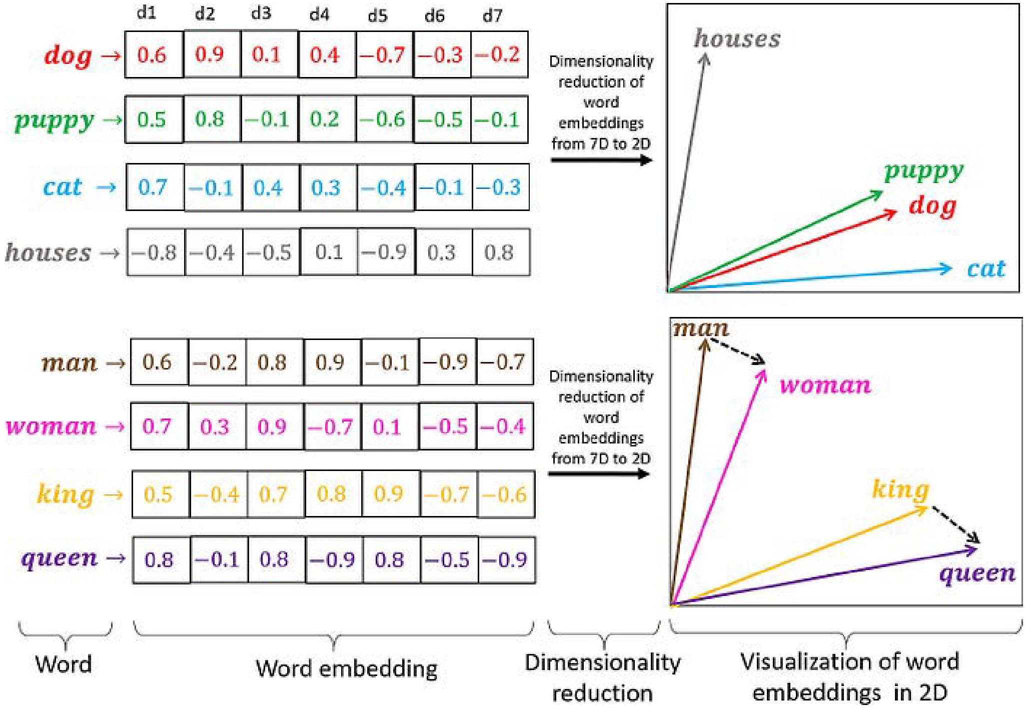
PCA and LDA

Principal Component Analysis and Linear Discriminant Analysis

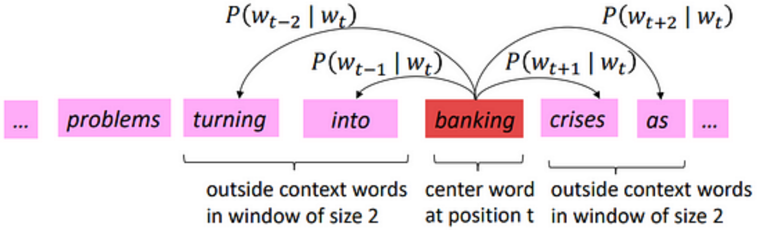


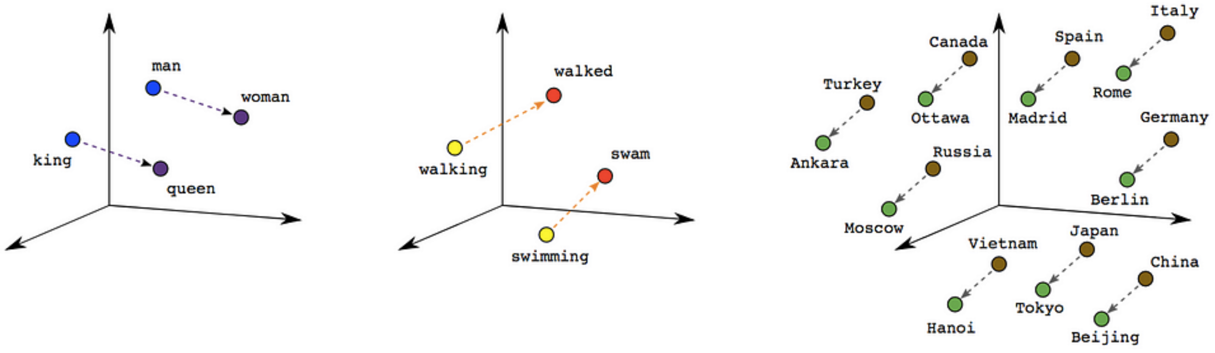
(Word) Embeddings





Example windows and process for computing $P(w_{t+j} \mid w_t)$





Male-Female Verb Tense Country-Capital

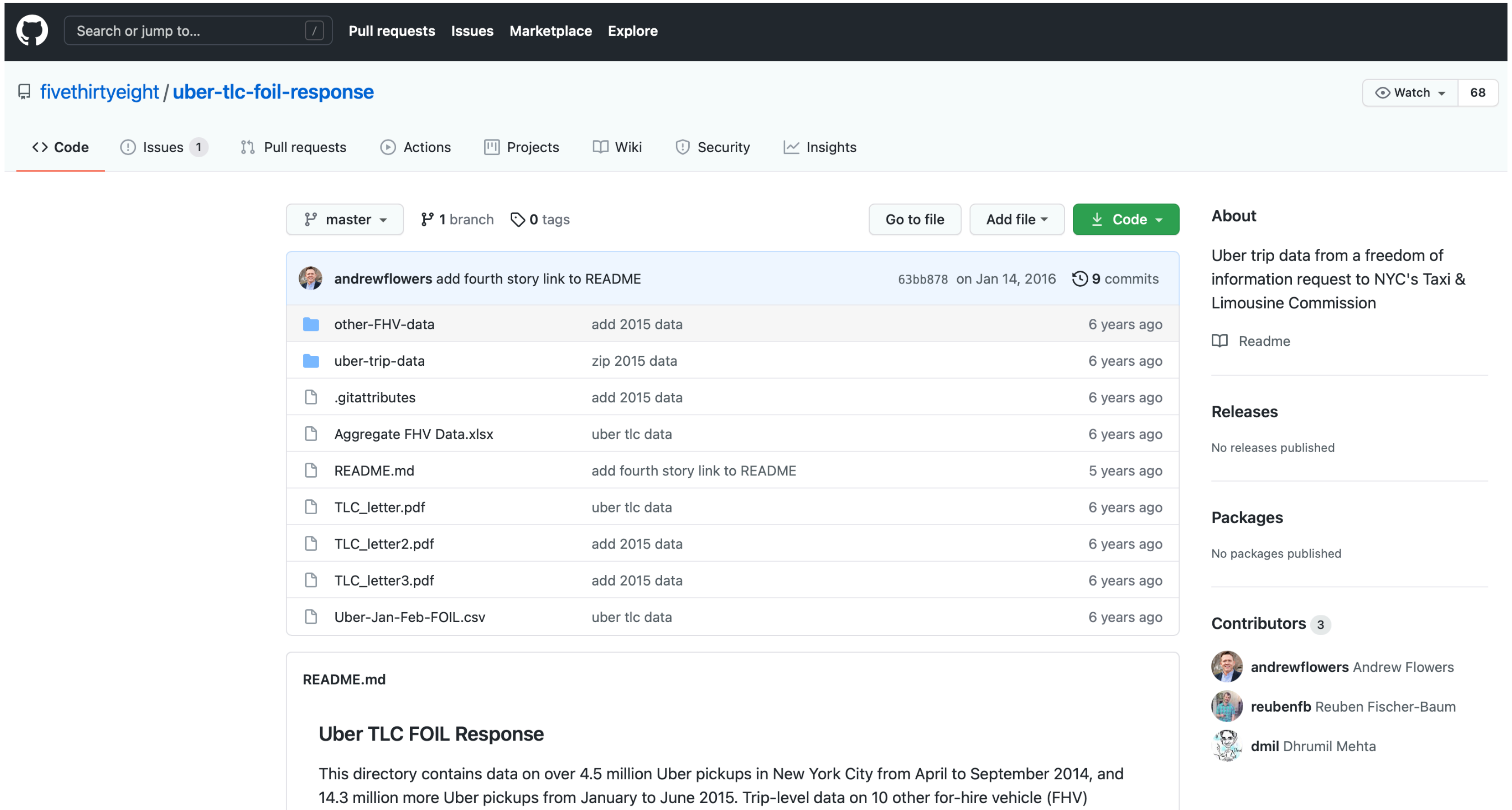
K-means Clustering in R

Use the the **`cluster` and `factoextra`** packages

Uber NYC pick up locations – here, the question is: *Do our NYC Uber pick up locations naturally fall into different groups?*

```
library(tidyverse)
library(cluster)    # clustering algorithms
library(factoextra) # clustering algorithms & visualization

apr14 <- read_csv("uber-raw-data-apr14.csv")
may14 <- read_csv("uber-raw-data-may14.csv")
jun14 <- read_csv("uber-raw-data-jun14.csv")
jul14 <- read_csv("uber-raw-data-jul14.csv")
aug14 <- read_csv("uber-raw-data-aug14.csv")
sep14 <- read_csv("uber-raw-data-sep14.csv")
```

Uber Data


Search or jump to...
Pull requests
Issues
Marketplace
Explore
fivethirtyeight / uber-tlc-foil-response
Watch 68
Code
Issues 1
Pull requests
Actions
Projects
Wiki
Security
Insights
master
1 branch
0 tags
Go to file
Add file
Code
andrewflowers add fourth story link to README
63bb878 on Jan 14, 2016
9 commits
other-FHV-data add 2015 data 6 years ago
uber-trip-data zip 2015 data 6 years ago
.gitattributes add 2015 data 6 years ago
Aggregate FHV Data.xlsx uber tlc data 6 years ago
README.md add fourth story link to README 5 years ago
TLC_letter.pdf uber tlc data 6 years ago
TLC_letter2.pdf add 2015 data 6 years ago
TLC_letter3.pdf add 2015 data 6 years ago
Uber-Jan-Feb-FOIL.csv uber tlc data 6 years ago
README.md
Uber TLC FOIL Response
This directory contains data on over 4.5 million Uber pickups in New York City from April to September 2014, and 14.3 million more Uber pickups from January to June 2015. Trip-level data on 10 other for-hire vehicle (FHV)
About
Uber trip data from a freedom of information request to NYC's Taxi & Limousine Commission
Readme
Releases
No releases published
Packages
No packages published
Contributors 3
andrewflowers Andrew Flowers
reubenfb Reuben Fischer-Baum
dmil Dhrumil Mehta

Join Data

```r
df <- bind_rows(…)
# Date and time of Uber pickup
# Latitude and Longitude of the Uber pickup
# Taxi and Limousine Commission (TLC) company code affiliated with the Uber pickup

df
```

A tibble: 4,534,327 x 4

Date/Time <chr>	Lat <dbl>	Lon <dbl>	Base <chr>
4/1/2014 0:11:00	40.7690	-73.9549	B02512
4/1/2014 0:17:00	40.7267	-74.0345	B02512
4/1/2014 0:21:00	40.7316	-73.9873	B02512
4/1/2014 0:28:00	40.7588	-73.9776	B02512
4/1/2014 0:33:00	40.7594	-73.9722	B02512
4/1/2014 0:33:00	40.7383	-74.0403	B02512
4/1/2014 0:39:00	40.7223	-73.9887	B02512
4/1/2014 0:45:00	40.7620	-73.9790	B02512
4/1/2014 0:55:00	40.7524	-73.9960	B02512
4/1/2014 1:01:00	40.7575	-73.9846	B02512

1–10 of 4,534,327 rows   Previous 1 2 3 4 5 6 … 100 Next

Base Code	Base Name
B02512	Unter
B02598	Hinter
B02617	Weiter
B02682	Schmecken
B02764	Danach-NY
B02765	Grun
B02835	Dreist
B02836	Drinnen

Splitting Data

Set a starting value so that results are reproducible
Draw random sample to run the analysis

```
set.seed(12L) # set a starting seed to be able to get reproducible results
# debate whether to partition data
# some say it is not needed because unsupervised learning is not
# concerned with prediction. Others say it is needed to avoid overfitting
# we will pull a random sample because the data is very large!

df1 <- sample_n(df, 50000)
```

Creating Clusters

Use **kmeans** to create the clusters and then save results back in the original data

```
# create cluster of pick up regions (columns 2 and 3)
clusters <- kmeans(df1[, 2:3], 5, nstart = 25) # create 5 clusters
# the Bronx, Brooklyn, Manhattan, Queens, and Staten Island
# 25 random sets to be chosen

# save the cluster number in the dataset as column 'borough'
df1$borough <- as.factor(clusters$cluster)
```

Inspecting Results

```
> clusters$size # cluster 5 has the most observations
[1] 22368   501  2714  1614 22803
```

```
# cluster structure
str(clusters)
# cluster: which cluster does this observation belong to?
# centers: a matrix of the center of each cluster
clusters$centers
# size: the number of observations in each cluster
clusters$size # cluster 5 has the most observations
```

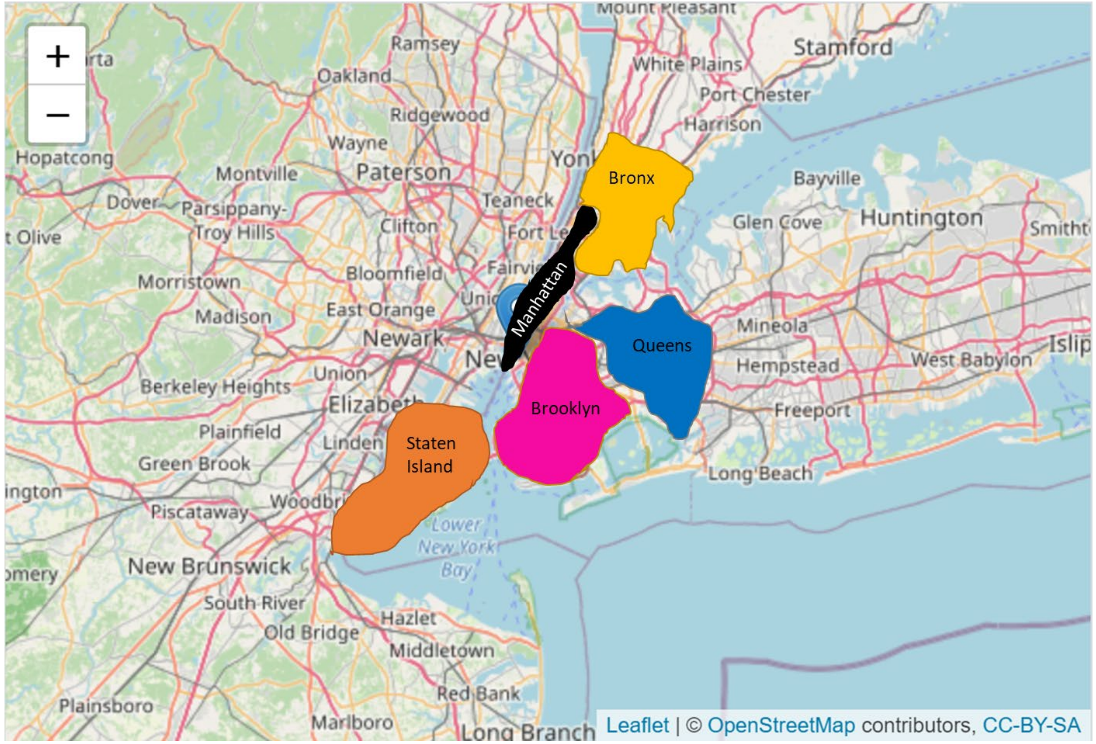
```
> clusters$centers
       Lat       Lon
1 40.71570 -73.98966
2 40.68990 -74.19979
3 40.79764 -73.88232
4 40.66620 -73.76581
5 40.76188 -73.97714
```

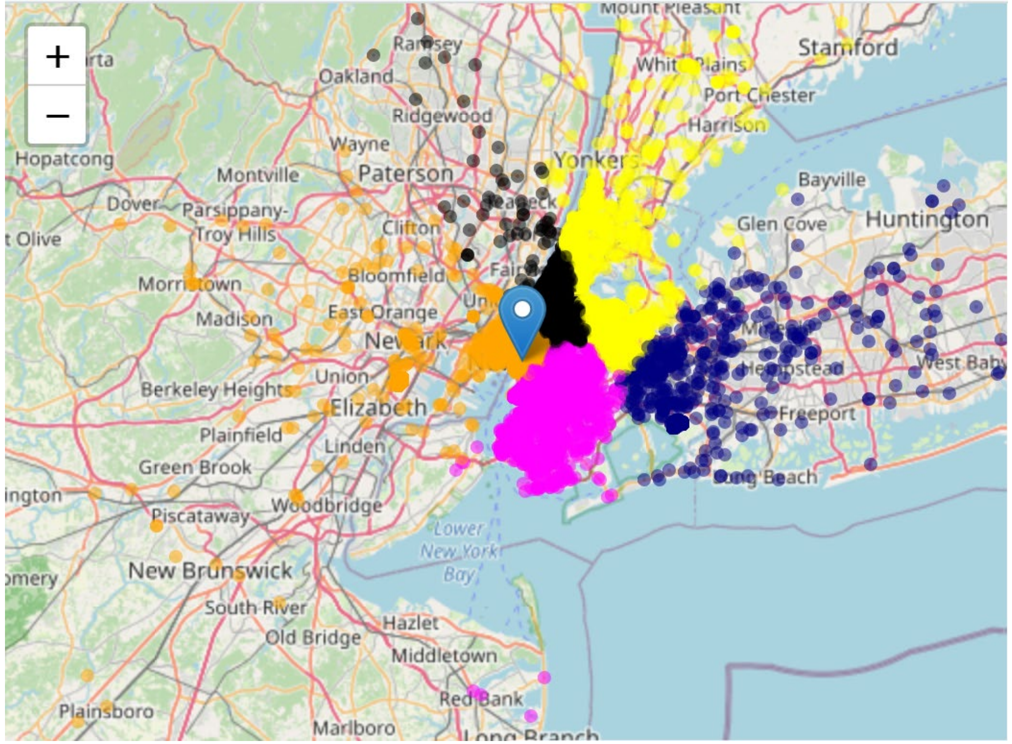
```
> str(clusters)
List of 9
 $ cluster     : int [1:50000] 1 1 1 5 5 3 5 1 5 1 ...
 $ centers     : num [1:5, 1:2] 40.7 40.7 40.8 40.7 40.8 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:5] "1" "2" "3" "4" ...
  .. ..$ : chr [1:2] "Lat" "Lon"
 $ totss       : num 241
 $ withinss    : num [1:5] 25.08 7.13 15.23 14.17 11.85
 $ tot.withinss: num 73.5
 $ betweenss   : num 167
 $ size        : int [1:5] 22368 501 2714 1614 22803
 $ iter        : int 3
 $ ifault      : int 0
 - attr(*, "class")= chr "kmeans"
```

Visualizing Results

```
# visualizing results
library(leaflet)
library(widgetframe)

pal <- colorFactor(c("magenta", "orange", "yellow", "navy", "black"), domain = c(1:5))

leaflet(data = df1) %>%
  addTiles() %>%
  addMarkers(lng = -74.0060, lat = 40.7128)

leaflet(data = df1) %>%
  addTiles() %>%
  addMarkers(lng = -74.0060, lat = 40.7128) %>%
  addCircleMarkers(label = ~borough, color = ~pal(borough), radius = 1)
```

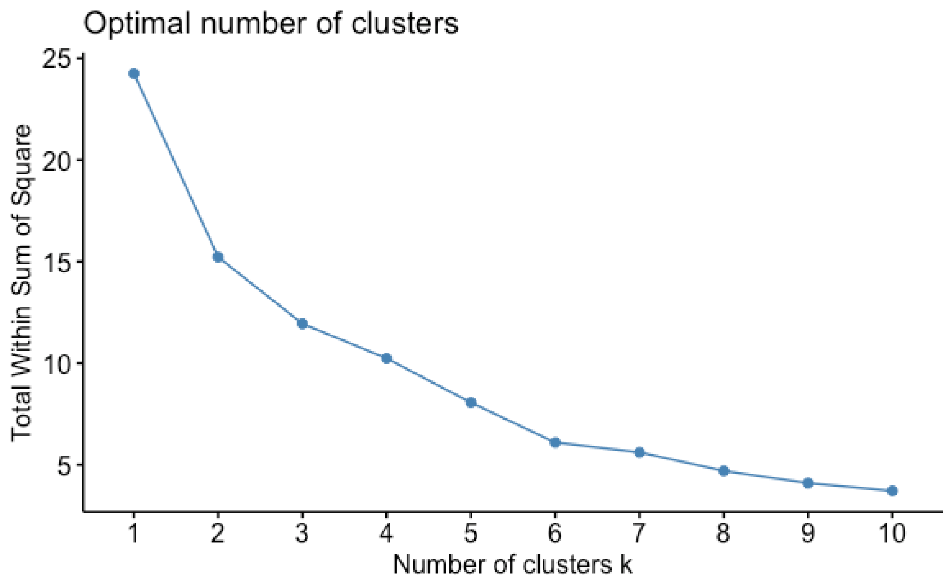
Visualizing Results





How many clusters?

```
# number of clusters?
df2 <- df1 %>% select(Lat, Lon) %>% sample_n(5000) # memory exhausted
fviz_nbclust(df2, kmeans, method = "wss")
# 2, 3, 4, 5, or 6? Tough to tell
```



Summary – Overfitting and Regularization

- Overfitting occurs when the model fits the training data too well that it does not generalize to the population

- Regularization is a useful way to address overfitting for when we have a lot of features and lasso is especially useful because of automatic feature selection

Summary – Unsupervised Learning

- K-means is a useful and very popular unsupervised learning technique for clustering data in *exploratory* analysis
- After random initialization, k-means does two things in an iterative way: 1) assigns points to cluster centroids and then 2) moves centroids around
- The choice of K is subjective and chosen a priori. Such choice can be very difficult – you can rely on the elbow method for guidance but also on your domain knowledge