CSE 4/587
Data Intensive Computing

Dr. Eric Mikida
epmikida@buffalo.edu
208 Capen Hall

Lec 06
Review of Models/Algorithms and Demo

Announcements and Feedback

- Project Phase 1 description should be released by tonight (will be announced on Piazza)

Recap

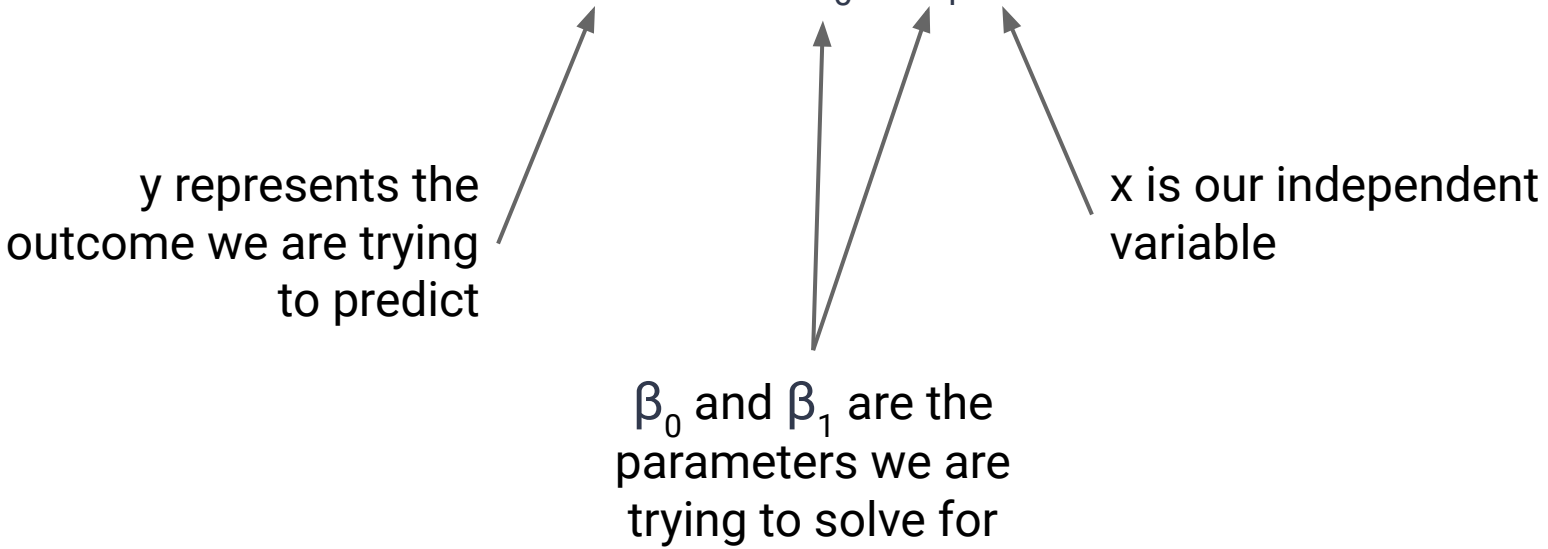
- So far we've looked at three different modeling algorithms
  - Linear regression
  - k-Nearest Neighbors
  - k-Means
- Each one has a different purpose and different factors consider as you develop your model/algorithm
- Today we will review each one, and demo it in Python

Linear Regression

- Attempts to model the relationship between two or more variables
  - If we can model the actual relationship, then our model can be used to **predict** outcomes not included in our dataset
- Linear regression is an example of a supervised algorithm: we know what the "right" answer is, and we use that to train the model
  - The "right" answer in this case is the observed value of an outcome variable for given values in our predictor variables

Linear Regression

- Specifically, we assume the underlying data is related in the real world by a function of the form: $y = f(x) = \beta_0 + \beta_1 x$



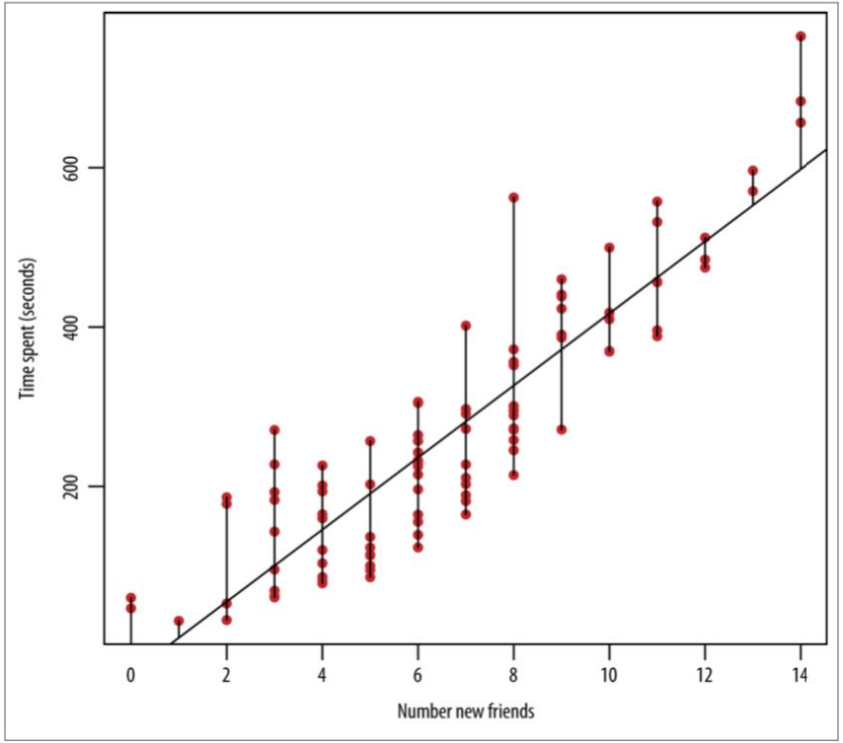
Linear Regression

Intuition

Each observed point has some vertical distance from a regression line.

The best line is the one that minimizes this distance.

In particular we minimize the square of the distance (least squares estimation).



Linear Regression - Example

- In Real Estate, what determines the sale price of a property?

Linear Regression - Example

- In Real Estate, what determines the sale price of a property?
  - Size (of the building and of the land?)
  - Location
  - Date of Sale
  - Age of the property
  - Type of property (commercial, residential, rental, etc)
  - Quality of the build
  - Amenities
  - etc...

Linear Regression - Developing a Model

- What data do we have? (and do we need to get more?)

Linear Regression - Developing a Model

- What data do we have? (and do we need to get more?)
  - Square footage: numeric, presumably bigger properties cost more

Linear Regression - Developing a Model

- What data do we have? (and do we need to get more?)
  - Square footage: numeric, presumably bigger properties cost more
  - Location: neighborhood name…non-numeric
    - Are there underlying factors at the core of this?
      - Crime Rate? Schools? Number of parks? etc…

Linear Regression - Developing a Model

- What data do we have? (and do we need to get more?)
  - Square footage: numeric, presumably bigger properties cost more
  - Location: neighborhood name…non-numeric
    - Are there underlying factors at the core of this?
      - Crime Rate? Schools? Number of parks? etc…
  - Property Type: class of building…non-numeric
    - Does it make sense to include all classes in our model? Maybe we want to model each class separately? Depends on our problem.

Linear Regression - Developing a Model

- What data do we have? (and do we need to get more?)
  - **Square footage: numeric, presumably bigger properties cost more**

To start, let's focus on square footage.

Thought experiment: What if square footage was the only factor in the sale price of a property? What might the model look like?

Demo in JupyterLab

Takeaways?

- The more thorough your cleaning and EDA, the easier the modeling process becomes
- Understand what you are modeling
  - If neighborhood has a large impact on sale price, we need to capture that in our model or use different models per neighborhood
- Sometimes the best answer is more data

k-Nearest Neighbors

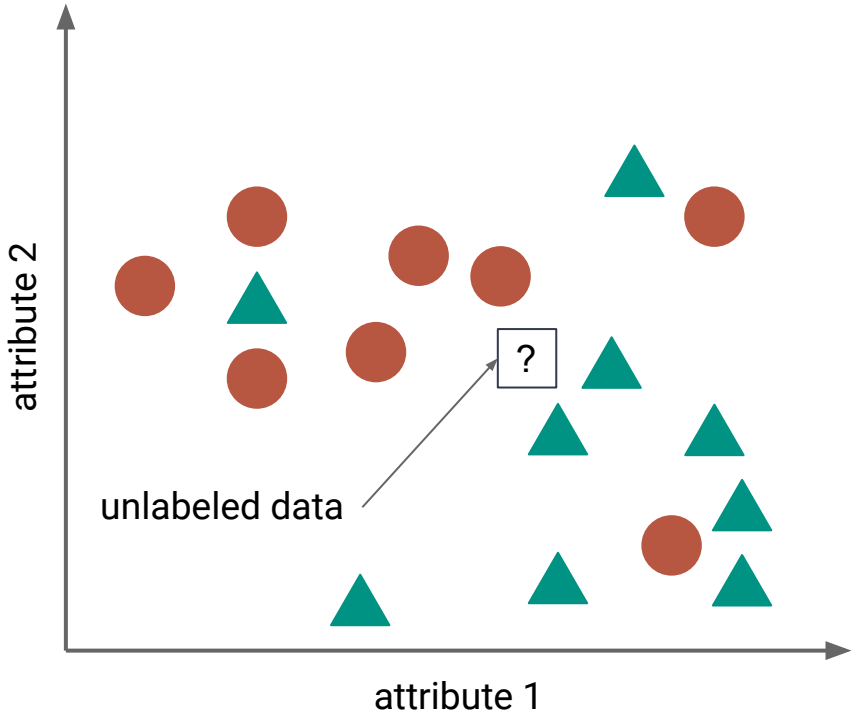
- Attempts to label data points based on training
  - No analog to a "model" of the real world
  - Used to **classify** new data points based on their similarity to already classified data points in our training dataset
- k-Nearest Neighbors is an example of a supervised algorithm
  - The "right" answer in this case is the labels of data points in our training dataset

k-Nearest Neighbors

Intuition:

For a given point, look at labels of the k "nearest" neighbors.

Label the unknown point with the most common label among these neighbors.



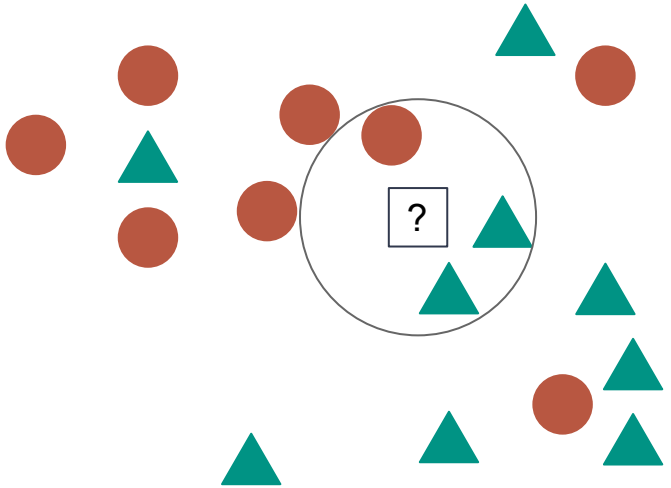
k-Nearest Neighbors

Intuition:

For a given point, look at labels of the k "nearest" neighbors.

Label the unknown point with the most common label among these neighbors.

k=3, label is "Green Triangle"



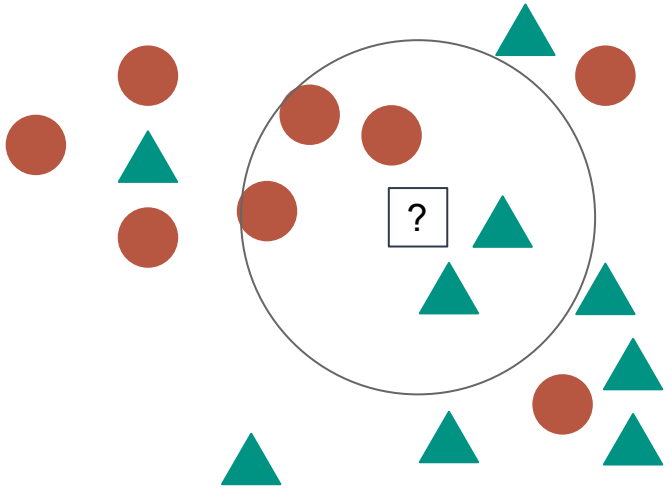
k-Nearest Neighbors

Intuition:

For a given point, look at labels of the k "nearest" neighbors.

Label the unknown point with the most common label among these neighbors.

k=5, label is "Red Circle"



k-Nearest Neighbors - Model Considerations

- What are the important factors we need to determine in order to fully define our model?

k-Nearest Neighbors - Model Considerations

- What are the important factors we need to determine in order to fully define our model?
  - How do we measure the "similarity" of two points?
    - Relatedly, how do we scale our data?
  - How do we evaluate our model's effectiveness?
  - What is the best choice for k?

k-Nearest Neighbors - Example

- Can we use characteristics of a given property to classify the neighborhood of the property? The type of property?
  - Let's use all of the numeric attributes we have: size, price, year built
  - *As an aside…does this make sense? It works as an example, but might not be a problem that makes sense to solve…*

Demo in JupyterLab

Takeaways?

- Scale can have a significant impact on your results
- Make sure what you are using in your model makes sense for the problem you are trying to solve
  - And if needed…get more data :)

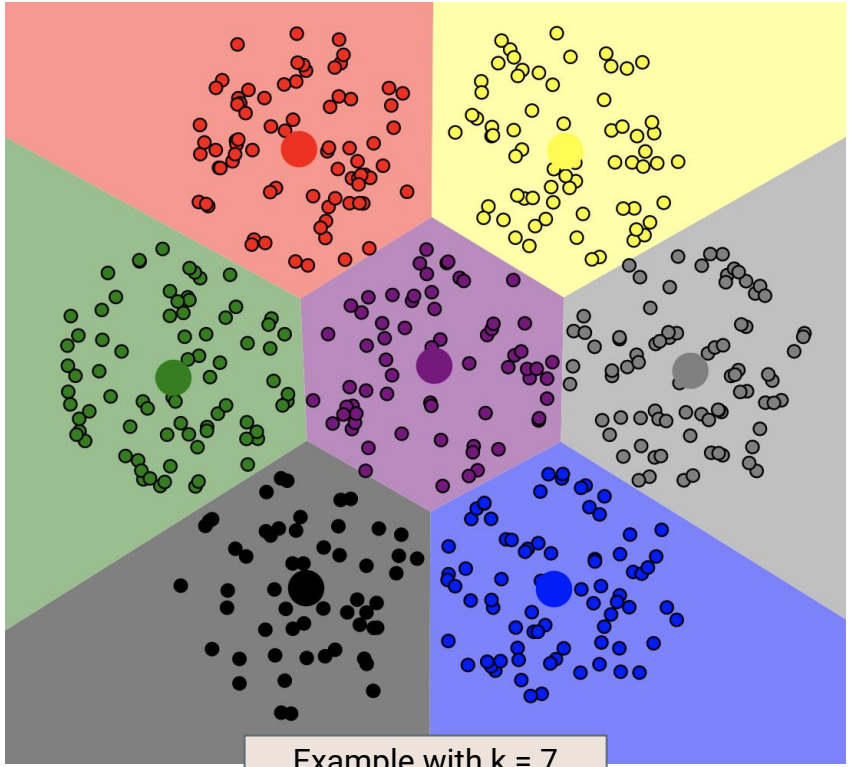
k-Means

- Attempts to **cluster** data into k groups based on similar attributes
  - No analog to a "model" of the real world, just looking for patterns in data
- k-Means is an example of an unsupervised algorithm: we don't have a notion of the "right" way to cluster the data to guide the algorithm

k-Means

Intuition

Cluster points to minimize distance between each point in the cluster and the center of the cluster.

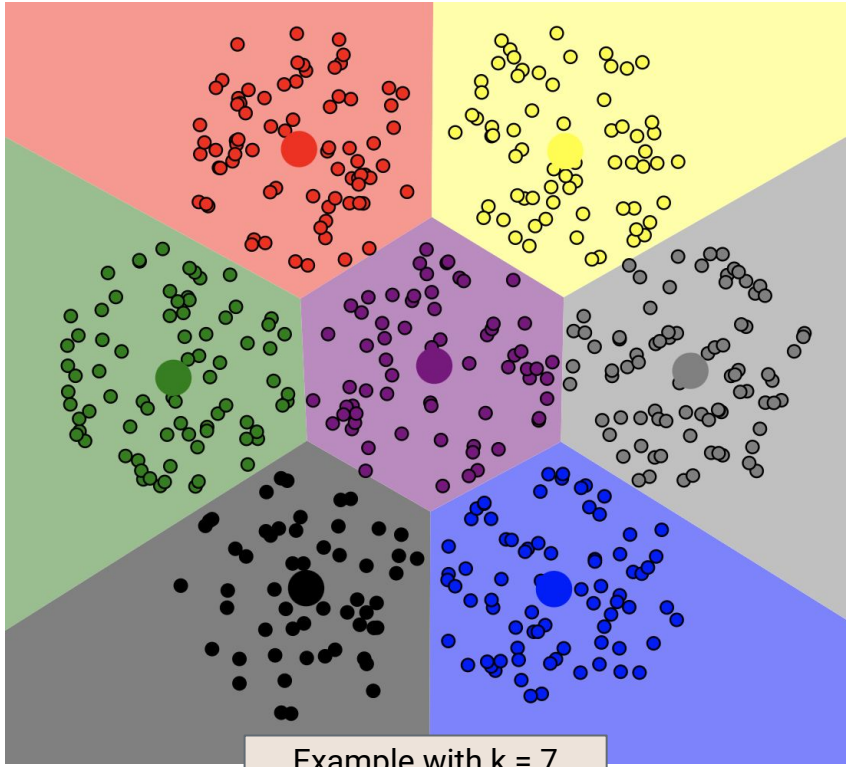


k-Means

Intuition

Cluster points to minimize distance between each point in the cluster and the center of the cluster.

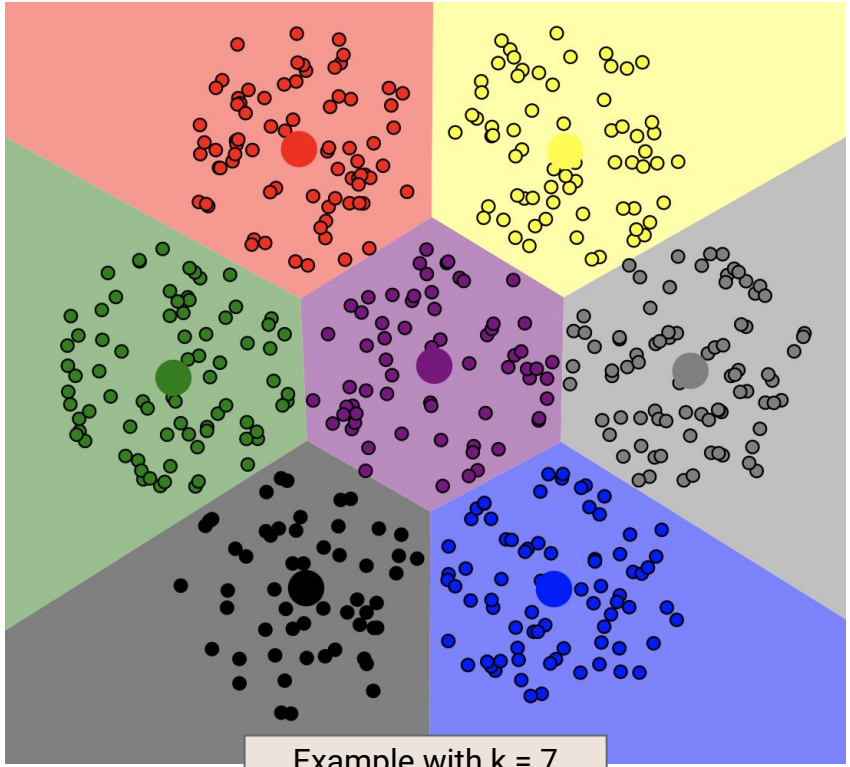
How do we find the centroids?



Example with k = 7

k-Means - Algorithm

1. Choose k centroids (randomly or via algorithm)
2. Until convergence:
   a. Assign each point to the closest centroid
   b. Move each centroid to the center of all points assigned to it



Example with k = 7

k-Means - Example

- How might we cluster properties?

k-Means - Example

- How might we cluster properties?
  - Perhaps we are trying to make policies/decisions based on size and age of a property
    - In a real scenario, may want to include more attributes, but for an example, 2D is easy to visualize
    - Motivation is going to based on problem statement and domain expertise

Demo in JupyterLab

Takeaways?

- Scale is an important part of a k-means model as well
- Results can be tricky to interpret
- Choice of k is also based on some sense of intuition/domain knowledge

Summary

- Three different algorithms, three **different** purposes
  - Important to pick the model that makes sense for your problem
  - Understanding the intuition behind the model, and the high level concepts at hands can help with this. But nothing replaces practice.
  - Do not underestimate the helpfulness of EDA
- Important to define all parts of your model. Parameter choices and even scaling data can have a large impact on results
- When in doubt, get more data
  - This example is on the small side…how can we apply these algorithms when size gets unruly

References

- Dataset from Doing Data Science Ch. 3
- Pandas tutorial: https://pandas.pydata.org/