topic 33: PCA

agenda:

1. unsupervised learning & what is PCA?
2. why PCA?
3. concept: how does PCA work?
4. evaluating PCA

unsupervised learning & PCA

- supervised learning is done to **predict** on **labelled data**
- unsupervised learning - there are no “labels”
  - i.e. clustering, PCA, LDA

dimensionality reduction

- PCA is a form of dimensionality reduction
- unsupervised because we only tell it **how many dimensions** to reduce to
- PCA reduces the dimensionality of the feature set into ***n* principal components** while maintaining its **variance**
- principal components are **linear combinations** of original features

why and when do we use PCA?

- the **curse of dimensionality**
  - as we have more features (columns), data points become more sparse
  - the distance between points gets greater, making it more difficult to implement some ML models
  - complexity: time and space complexity

- when to use PCA:
  - when you have a lot of continuous (not categorical) features!
  - best for clustering, or computationally-heavy ML algos like SVM

how does PCA work?

- linear algebra!
  - mathematically, **Principal Components** are found through doing **Eigendecomposition** of the **Covariance Matrix**

1. Recenter your data such that the means of each feature are 0
2. Get the covariance matrix (in Pandas: df.corr())
3. Get the **eigenvectors** of the covariance matrix
4. Sort the eigenvectors
5. Multiply the eigenvectors by the recentered data (for 2 PCs, multiply by the first two eigenvectors)

evaluating PCA

- how many Principal Components is best for your dataset?