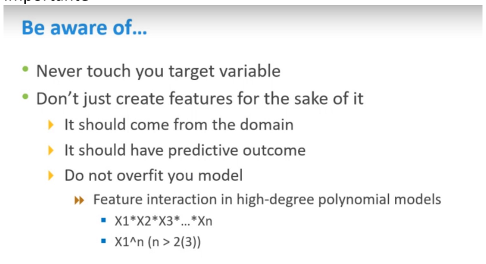
Lec 5

3 октября 2024 г. 19:02

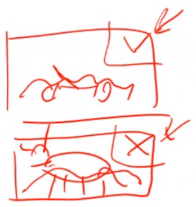
1. Distribution of data can tell us a lot which model should be used
2. ...
3. How to map some color information to model to predict price of a car? Category -> numeric feature
4. The naïve approach is to use come color model RGB or anything else, but it have very little to do with car price.
5. Usually it is solved by adding binary vars like isRed, isYellow and etc. But how effective is it?
6. ...
7. Picture? Huh, it's just pixels. But our features are not directly related to pixels. For example, we need to recognize car number on picture or an animal, but these features are not really about pixels. Feature engeneering is important
8. Importante



Be aware of…

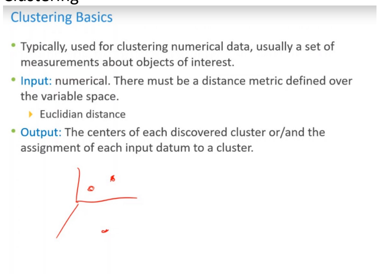
- Never touch you target variable
- Don't just create features for the sake of it
  - It should come from the domain
  - It should have predictive outcome
  - Do not overfit you model
    - Feature interaction in high-degree polynomial models
      - X1*X2*X3*…*Xn
      - X1^n (n > 2(3))

Don't add unnnecessery features. It can lead to overtraining.
You should not allow model to use data not from domain - use target value in dataset.
Example:



Tank system learns on cow/weapon images. But with check or cross in the corner. On the battlefield this system won't work, because it won't see any crosses and checks in the corner.

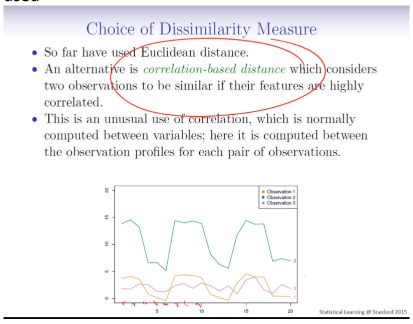
9. Clustering



Clustering Basics

- Typically, used for clustering numerical data, usually a set of measurements about objects of interest.
- Input: numerical. There must be a distance metric defined over the variable space.
  - Euclidian distance
- Output: The centers of each discovered cluster or/and the assignment of each input datum to a cluster.

In some space N looking for dots(vectors) which are similar and grouping them. But euclidian metric not always the best choice. Example where some correlation-based distance should be used

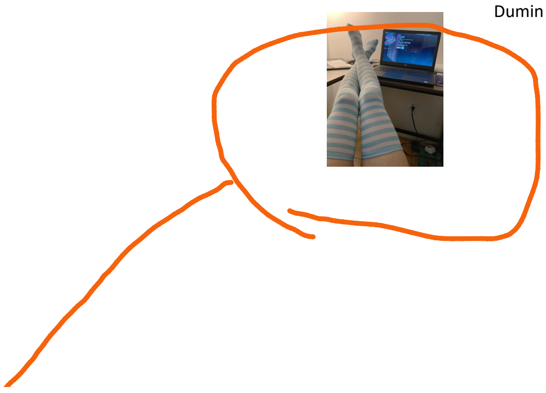


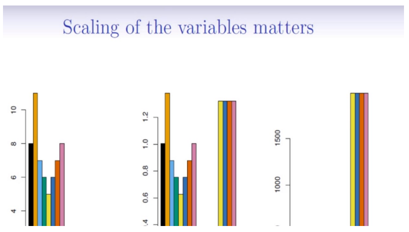
Choice of Dissimilarity Measure

- So far have used Euclidean distance.
- An alternative is *correlation-based distance* which considers two observations to be similar if their features are highly correlated.
- This is an unusual use of correlation, which is normally computed between variables; here it is computed between the observation profiles for each pair of observations.

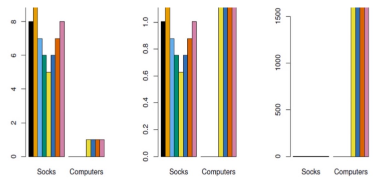
Statistical Learning @ Stanford 2015

Dumin wtf

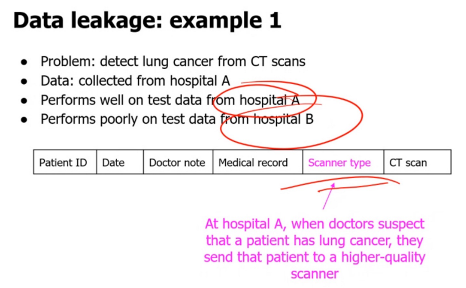




Scaling of the variables matters



How to clusterize? By items? By price? Somehow else? Depends on case we are working on. Mb Logistics? Perhaps items are more suitable.



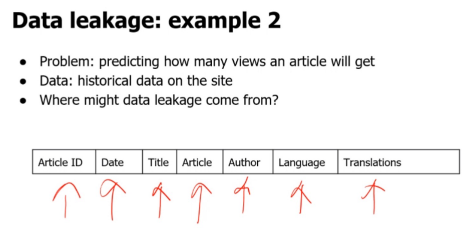
Data leakage: example 1

- Problem: detect lung cancer from CT scans
- Data: collected from hospital A
- Performs well on test data from hospital A
- Performs poorly on test data from hospital B

Patient ID	Date	Doctor note	Medical record	Scanner type	CT scan

At hospital A, when doctors suspect that a patient has lung cancer, they send that patient to a higher-quality scanner

Cause when patient is almost prooved to have lung cancer he get higher-quality scanner expertise to get more details about desease.



Data leakage: example 2

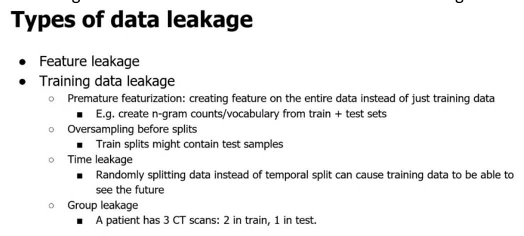
- Problem: predicting how many views an article will get
- Data: historical data on the site
- Where might data leakage come from?

Article ID	Date	Title	Article	Author	Language	Translations

Translations - cause more popular articles get more translations. Leak!

10. How to fight feature leaks? The best variant is to refuse using features that cause leaks.

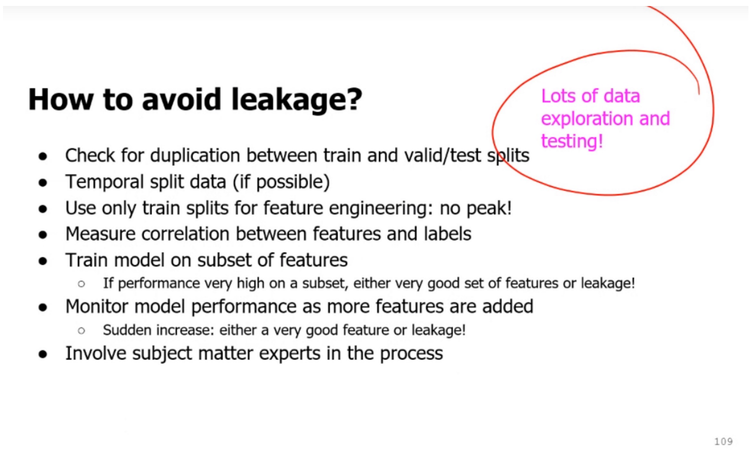
11.



Types of data leakage

- Feature leakage
- Training data leakage
  - Premature featurization: creating feature on the entire data instead of just training data
    - E.g. create n-gram counts/vocabulary from train + test sets
  - Oversampling before splits
    - Train splits might contain test samples
  - Time leakage
    - Randomly splitting data instead of temporal split can cause training data to be able to see the future
  - Group leakage
    - A patient has 3 CT scans: 2 in train, 1 in test.

12.



How to avoid leakage?

Lots of data exploration and testing!

- Check for duplication between train and valid/test splits
- Temporal split data (if possible)
- Use only train splits for feature engineering: no peak!
- Measure correlation between features and labels
- Train model on subset of features
  - If performance very high on a subset, either very good set of features or leakage!
- Monitor model performance as more features are added
  - Sudden increase: either a very good feature or leakage!
- Involve subject matter experts in the process

Feature engineering: the more the better?

- Adding more features tends to improve model performance
- Overfitting
- Too many features makes your model difficult to maintain
13. - The more feature engineering you do, the more chance for labels to leak
    - Feature A doesn’t cause leakage, but feature A + feature B can cause leakage