More Subtle Examples of Data Leakage

- **Prediction target: will user stay on a site, or leave?**
  - *Giveaway feature: total session length, based on information about future page visits*
- **Predicting if a user on a financial site is likely to open an account**
  - *An account number field that's only filled in once the user does open an account.*
- **Diagnostic test to predict a medical condition**
  - *The existing patient dataset contains a binary variable that happens to mark whether they had surgery for that condition.*
  - *Combinations of missing diagnosis codes that are not be available while the patient's condition was still being studied.*
  - *The patient ID could contain information about specific diagnosis paths (e.g. for routine visit vs specialist).*
- **Any of these leaked features is highly predictive of the target, but not legitimately available at the time prediction needs to be done.**

Other Examples of Data Leakage

Leakage in training data:

- Performing data preprocessing using parameters or results from analyzing the entire dataset: Normalizing and rescaling, detecting and removing outliers, estimating missing values, feature selection.
- Time-series datasets: using records from the future when computing features for the current prediction.
- Errors in data values/gathering or missing variable indicators (e.g. the special value 999) can encode information about missing data that reveals information about the future.

Leakage in features:

- Removing variables that are not legitimate without also removing variables that encode the same or related information (e.g. diagnosis info may still exist in patient ID).
- Reversing of intentional randomization or anonymization that reveals specific information about e.g. users not legitimately available in actual use.

Any of the above could be present in any external data joined to the training set.

Detecting Data Leakage

- **Before building the model**
  - *Exploratory data analysis to find surprises in the data*
  - *Are there features very highly correlated with the target value?*
- **After building the model**
  - *Look for surprising feature behavior in the fitted model.*
  - *Are there features with very high weights, or high information gain?*
  - *Simple rule-based models like decision trees can help with features like account numbers, patient IDs*
  - *Is overall model performance surprisingly good compared to known results on the same dataset, or for similar problems on similar datasets?*
- **Limited real-world deployment of the trained model**
  - *Potentially expensive in terms of development time, but more realistic*
  - *Is the trained model generalizing well to new data?*

Minimizing Data Leakage

- **Perform data preparation within each cross-validation fold separately**
  - *Scale/normalize data, perform feature selection, etc. within each fold separately, not using the entire dataset.*
  - *For any such parameters estimated on the training data, you must use those same parameters to prepare data on the corresponding held-out test fold.*
- **With time series data, use a timestamp cutoff**
  - *The cutoff value is set to the specific time point where prediction is to occur using current and past records.*
  - *Using a cutoff time will make sure you aren't accessing any data records that were gathered after the prediction time, i.e. in the future.*
- **Before any work with a new dataset, split off a final test validation dataset**
  - *… if you have enough data*
  - *Use this final test dataset as the very last step in your validation*
  - *Helps to check the true generalization performance of any trained models*