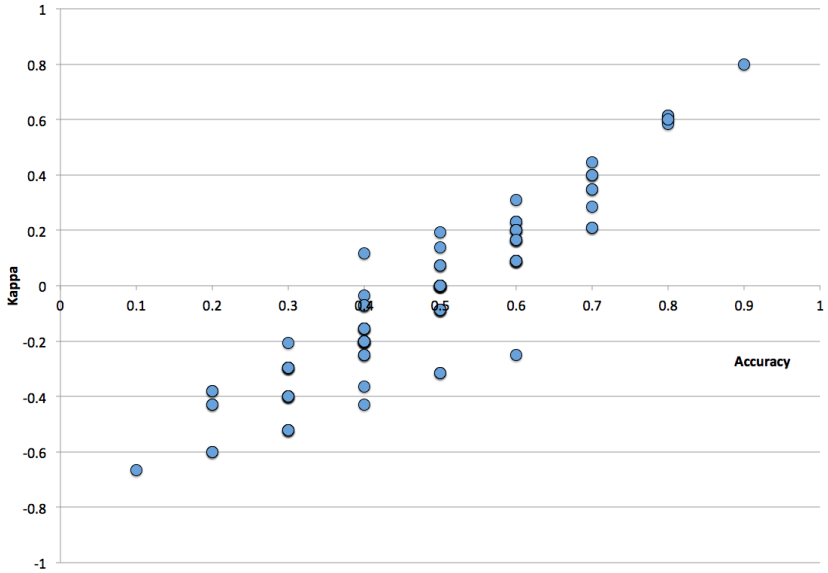
Cross Validation

Machine Learning With Python

Cross Validation

Basic idea is simple:

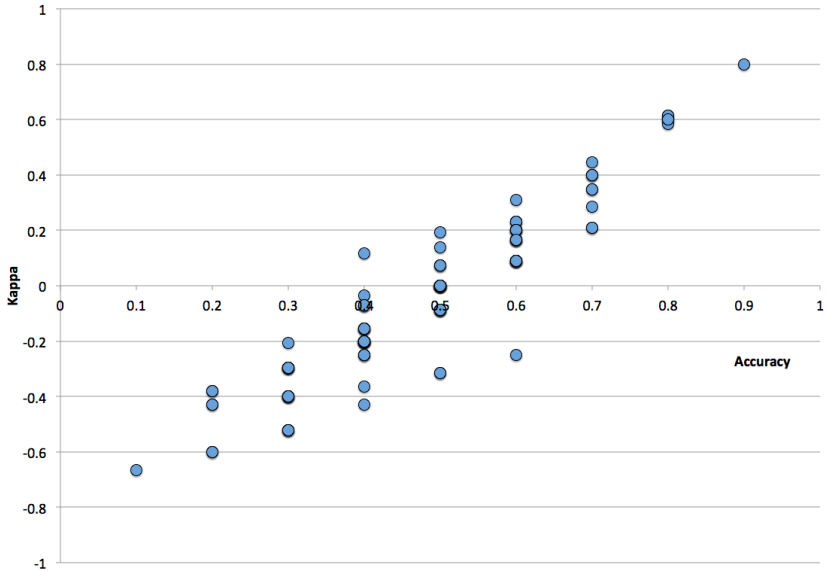
- Separate your data into two groups, a “test” set and a “training” set
- Test set is not touched until the very end, and is only used for final model evaluation
- Training set is used for fine-tuning and evaluating your model



Cross Validation

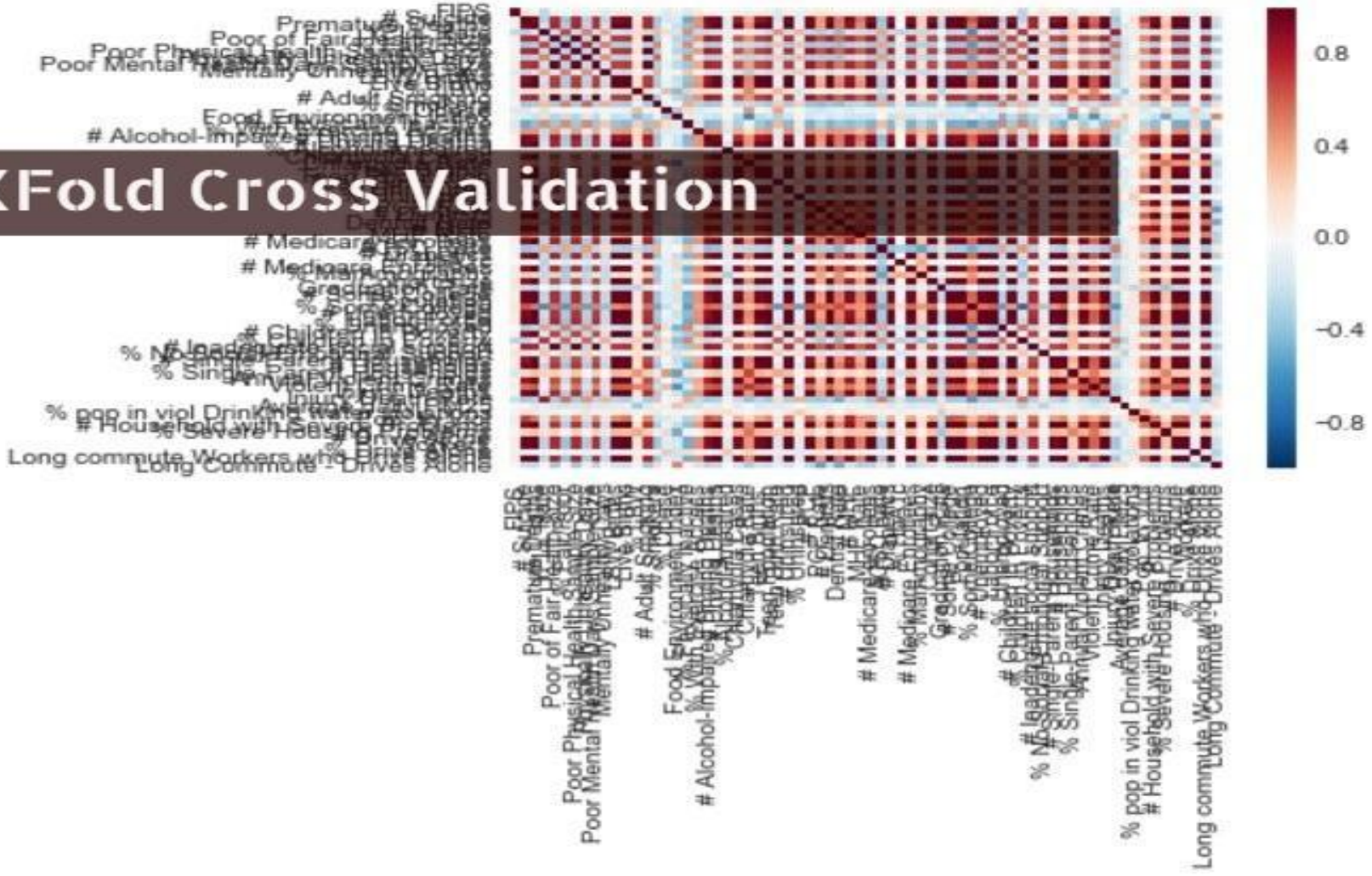
Additional step is to add a validation set:

- Test-set within the test-set
- Used to compare results when doing cross validation
- Meant to be a portion of your training set that resembles the test set



How Do We Know How Our Model Performs Under A Variety of Conditions?

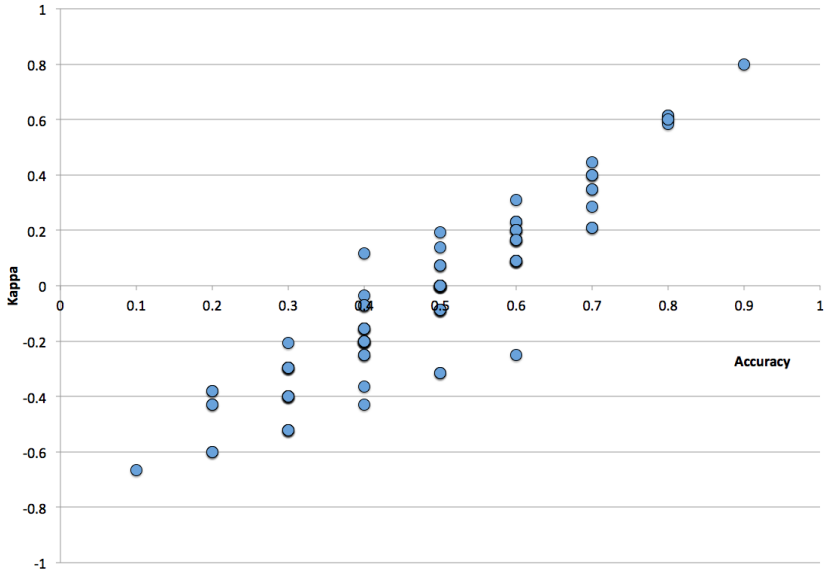
KFold Cross Validation

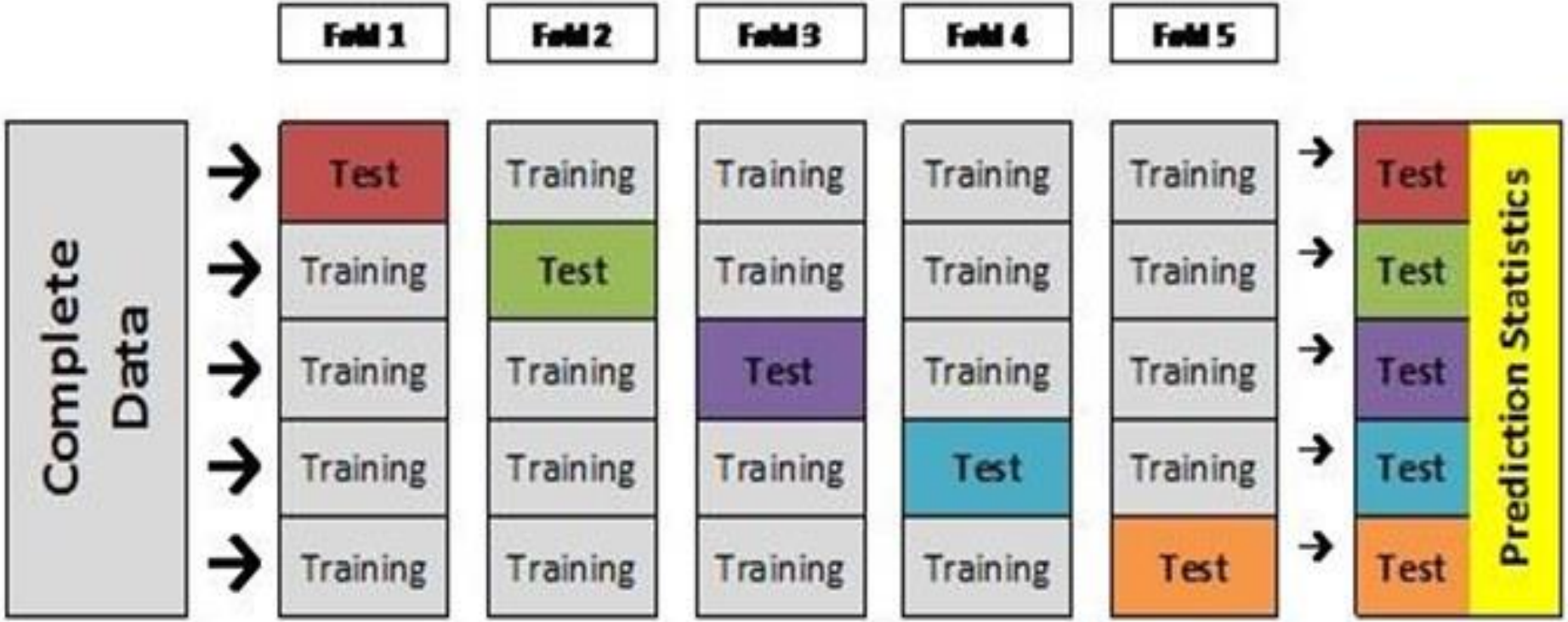


Cross Validation

Kfold Cross Validation:

- More thorough way of cross validation
- Split training set into K different groups
- For k rounds, train your data on K-1 groups, and score it on the one remaining set
- Is only used on the *training set*

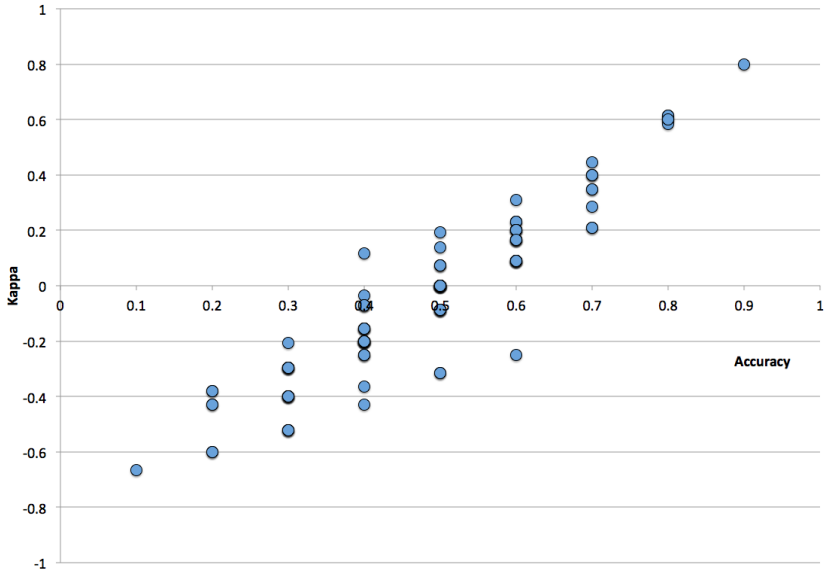




KFold

Key ideas:

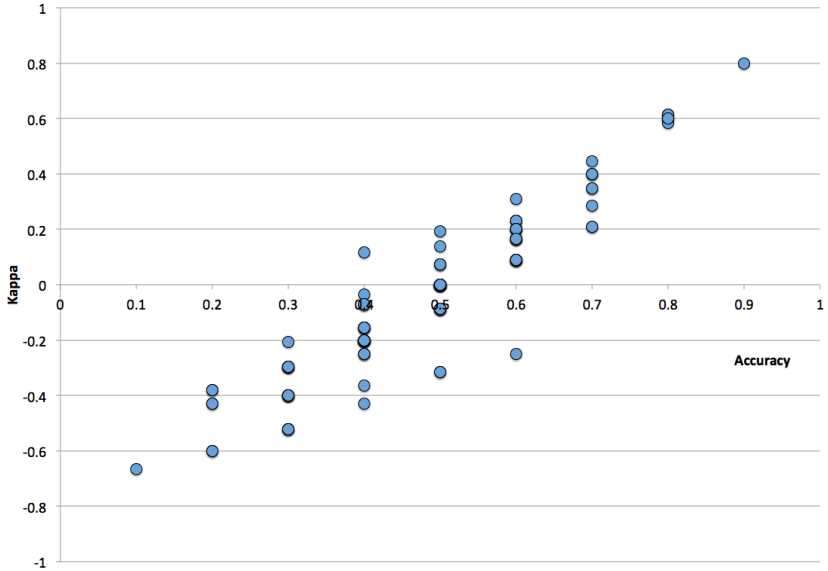
- *Every* row in the training set is used for both training and validation
- But *never* at the same time
- Different validation scores gives you an ability to see how model performs under different circumstances



Cross Validation

Key idea about cross validation:

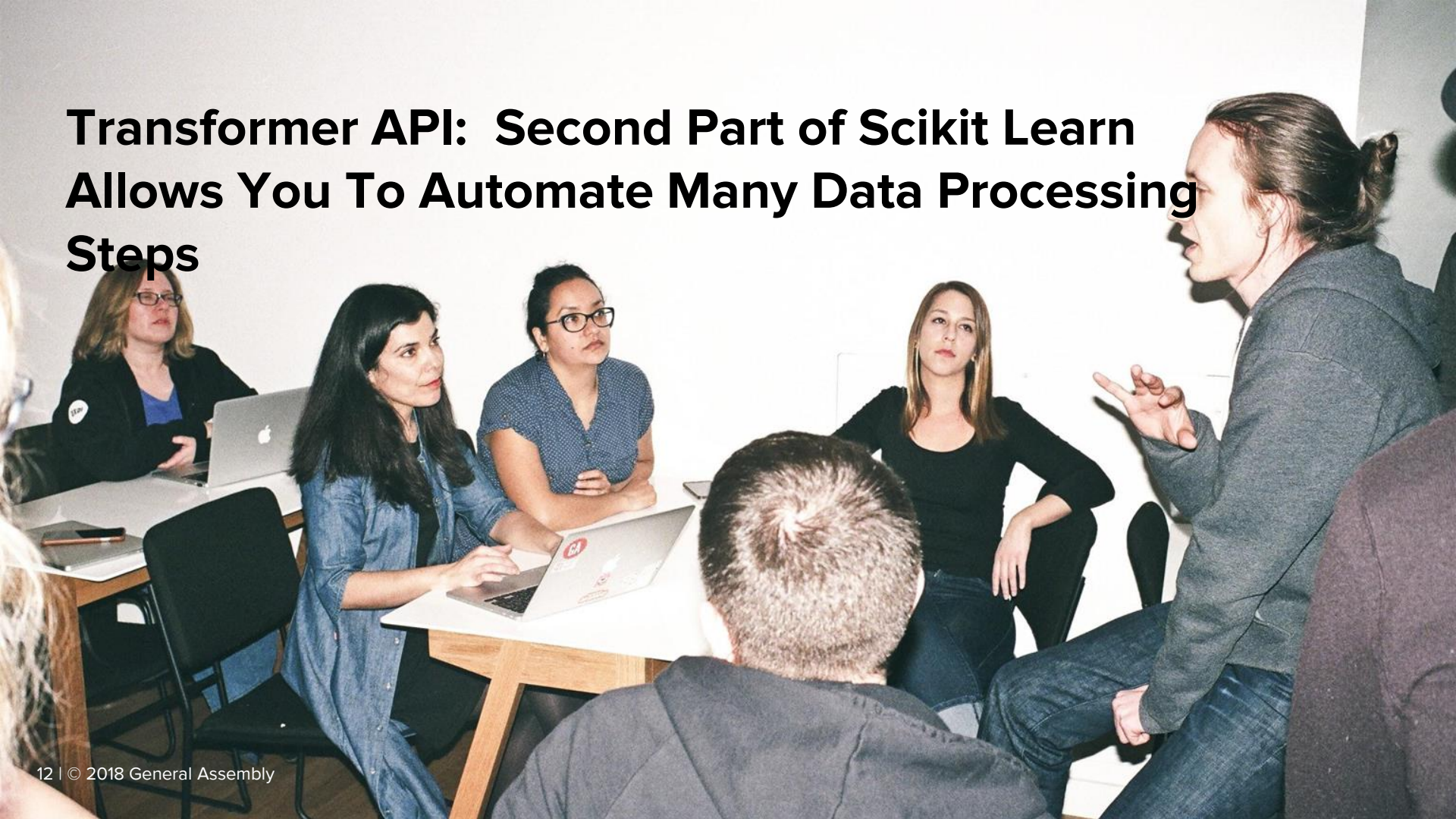
- Allows you to compare changes between models
- Ie, your cross validated score gives you a benchmark to use to determine whether or not you're making progress or not



Pipelines

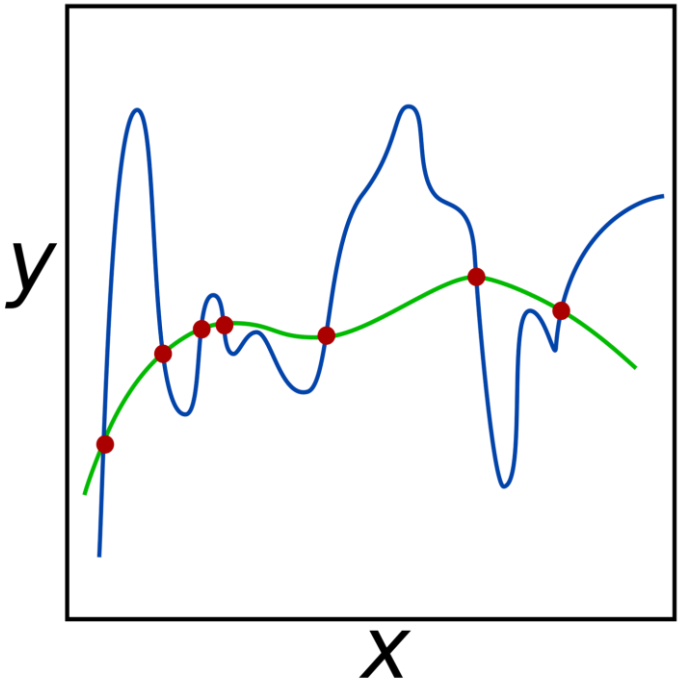
Observation: Data Handling And Prep Is A Pain

Transformer API: Second Part of Scikit Learn Allows You To Automate Many Data Processing Steps



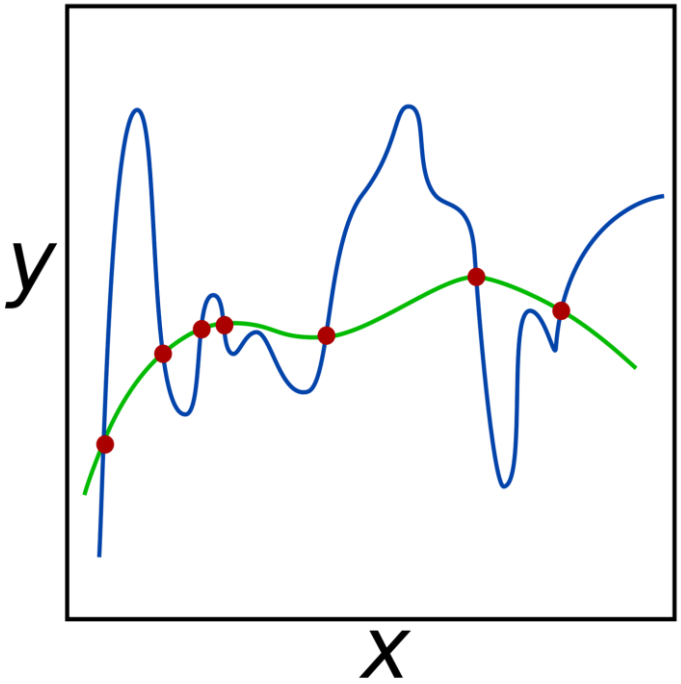
Major Function Calls

- **Fit:** Stores values in dataset necessary to do transformation
- **Transform:** Processes dataset in appropriate way
- **Fit_transform:** Does both steps in one swoop
- **Inverse_transform:** Restores dataset back to its original form



Key Benefits

- Allows you to separate concerns with training and test set processing more easily
- Multiple transformers can be chained together using *pipelines*
- You can incorporate external libraries into pipelines to give yourself a variety of options for processing data
- Pipelines can include *estimators,* to give yourself a succinct, end-to-end solution for processing data



Case In Point: Category Encoders Library

Category Encoders

- Under appreciated library of the python ecosystem
- Easiest way to encode categorical columns on training and test set
- Is designed to take pandas df's in, and put out pandas df's -- unlike scikit learn
- Contains a number of useful ways to encode categorical columns beyond ordinal & onehot -- useful for when you have columns with high amounts of unique values