



# Stacking, Pipelines and Model in Production



# Agenda

# Discussion Flow

- Stacking: Combining Multiple Models
- Python Pipelines : Structured approach to model deployment
- Pickle : Saving models for deployment
- Flask API : Models to prediction APIs in Production

# Stacking

# Why do we need complex algorithms

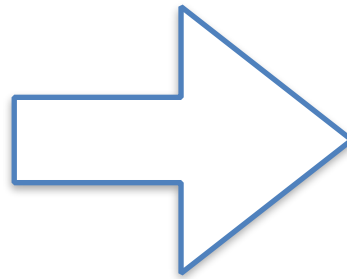
- When are linear models not enough ?
- Variable transformations
- Complex Algos : Capture complex non-linear patterns

# Complex algos as data transformers

- A complex algo function  $f$ , models the trends in data  $X$ , such that  $y = f(x) + e$
- Different algos ( randomforest , boosting machines, SVM) capture different patterns in the data
- So far we tested many and selected just one
- We can consider  $f$  to be a data transformation algorithm such that result ( prediction) is a linearly related to  $y$
- Now that we have complex transformations of the data, we can use these as inputs to our linear model

# Catch : Out of sample predictions

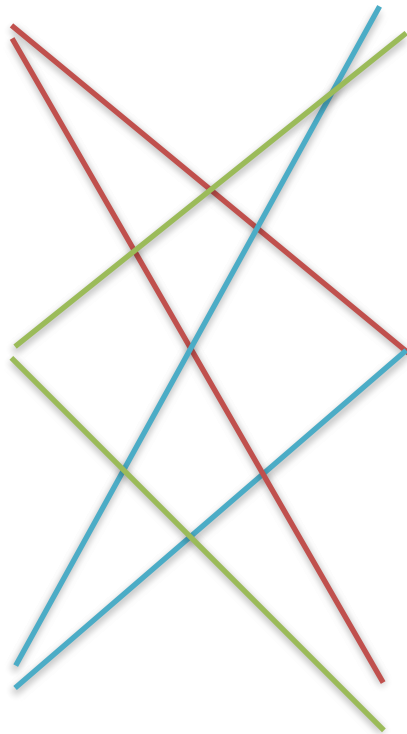
$x1$	$x2$	$x3$	....	$xn$	$y$



$f1$	$f2$	$f3$	$y$

# Cross Sampling to Rescue

$x_1$	$x_2$	$x_3$	....	$x_n$	$y$

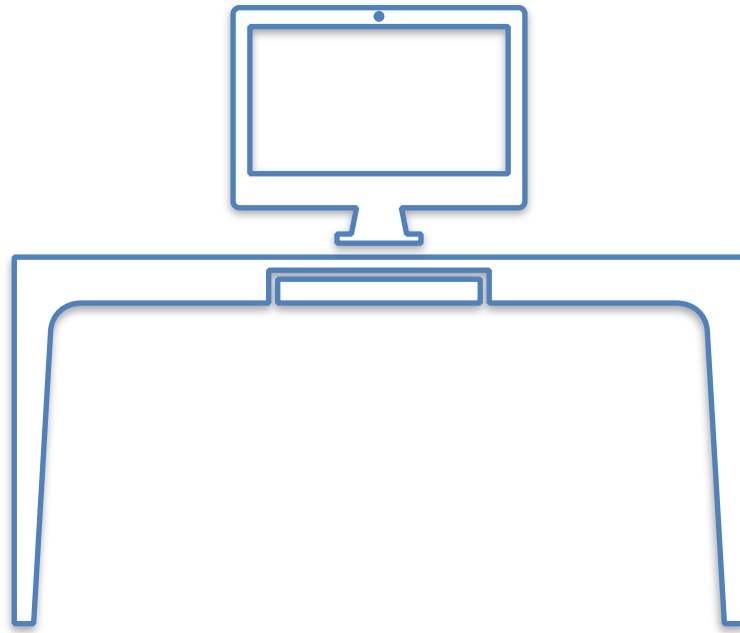


$f_1$	$f_2$	$f_3$	$y$

$$y = \beta_1 * f_1 + \beta_2 * f_2 + \beta_3 * f_3 + e$$



# Lets see it in action in Python

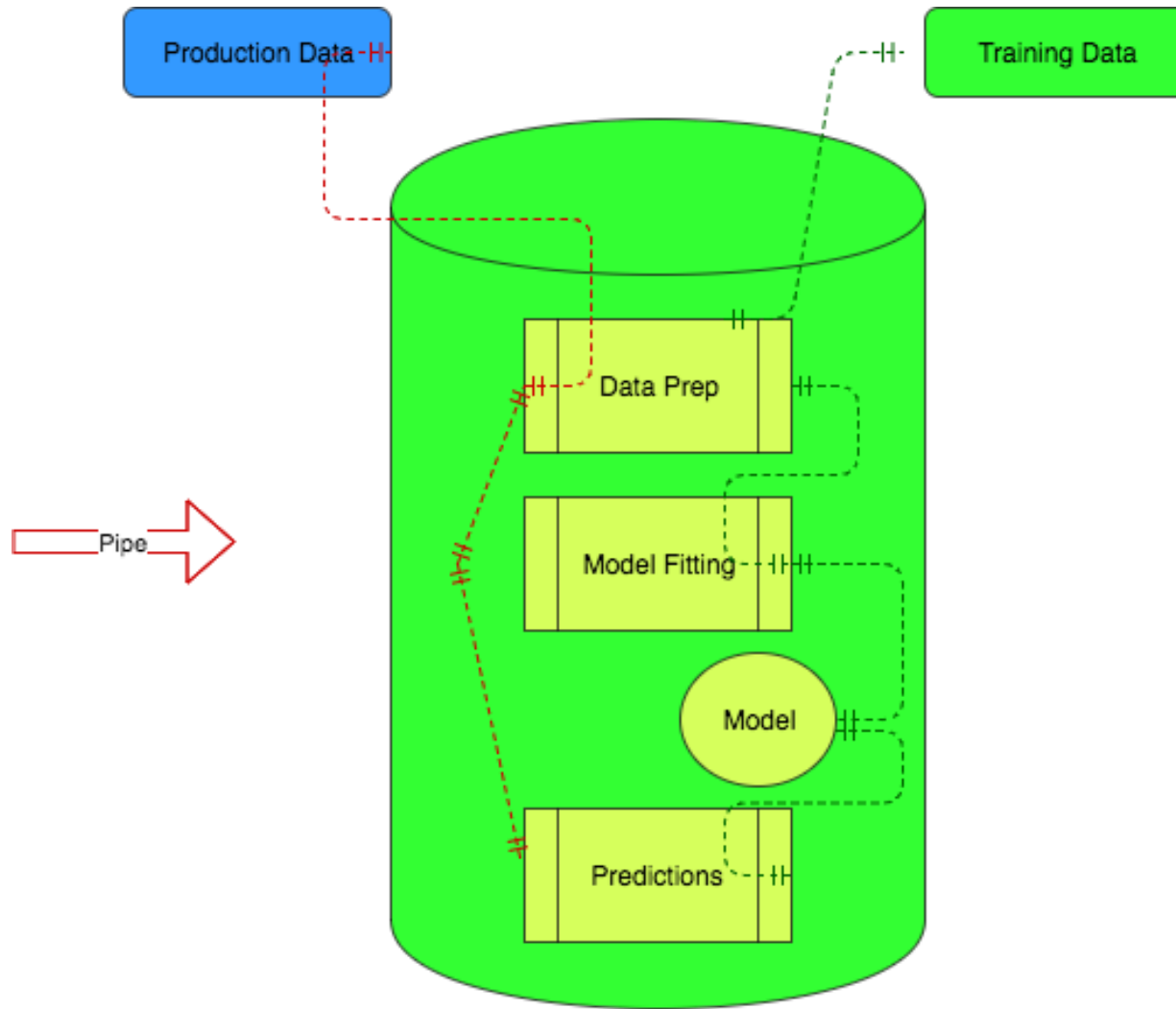


# Python Pipelines

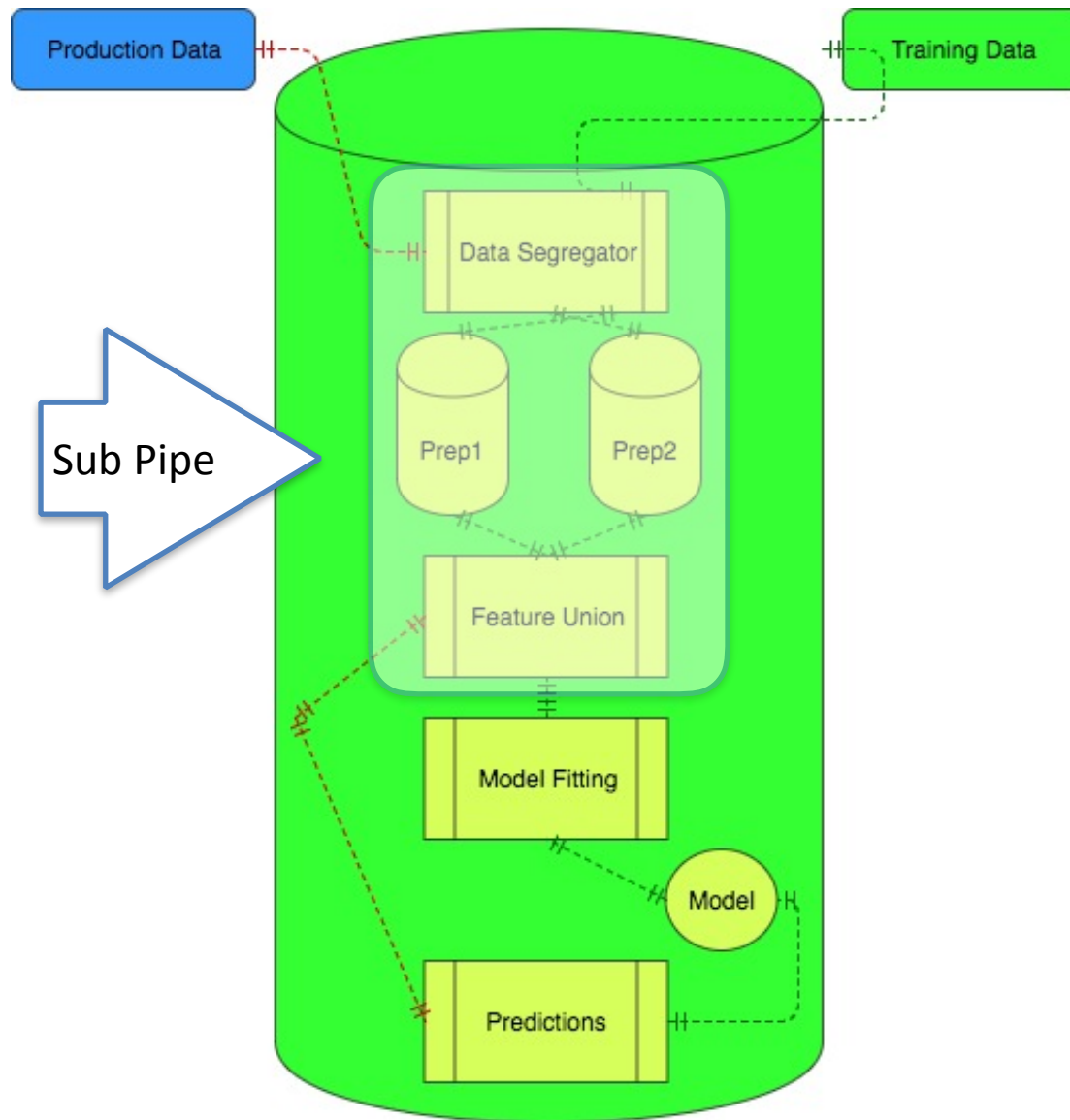
# Current Approach

- Combine training and production data for data prep
- Or Copy the multiple lines of data prep step and make the production data go through it
- Scope of manual error grows
- Data processing steps are not reusable ( code again for new problem )

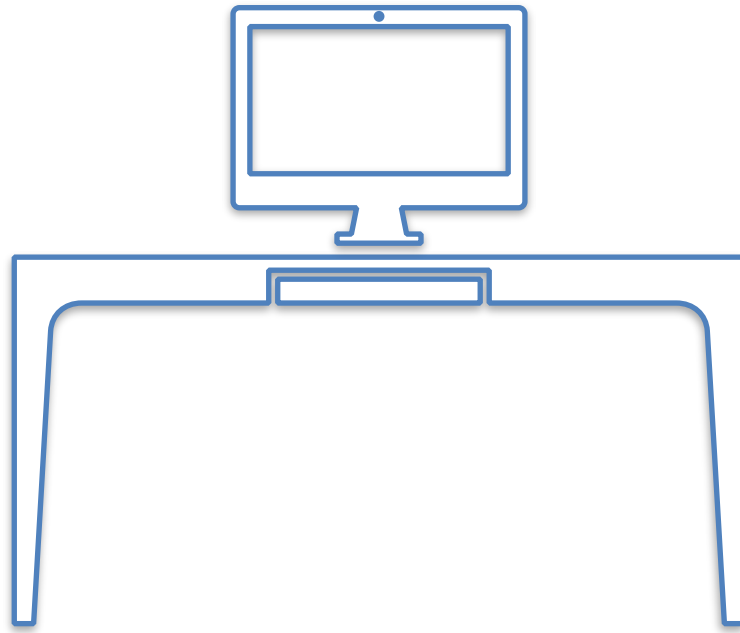
# Simple Pipeline



# Pipeline with feature union



# Lets see it in action in Python



# Model In Production

# Requirement For putting model in production

- Model building machines and production servers are separate
- Models can be saved and passed on to server
- No need to build model on server , they can be simply loaded
- Business process should be able to connect with server and get results to make decisions



# Lets see it in action in Python

