DECISION PATTERNS

Time Series Analysis and Forecasting

Practical Machine Learning (with R)

UC Berkeley

Spring 2016

Topics

- Administrativa
  - Role Call
  - Assignments due to github
- Review/Expectations
  - Readings
    - **APM** Chapter 8.6 and 8.8
    - **APM** Chapter 14.8
    - **APM** Chapter 7.1 & 7.3 "Non-Linear Regression Models"
    - **APM** Chapter 13.2 & 13.4 "Non-Linear Classification Models"
  - Previous Lecture
- New Topics

REVIEW AND EXPECTATIONS

IMPROVING MODELS

Two Big Ideas

- **Wisdom of the crowds**
  It is better to make estimates from multiple models (***ensembles***) than individual models
  - Better predictions
  - Lower variance for the same model

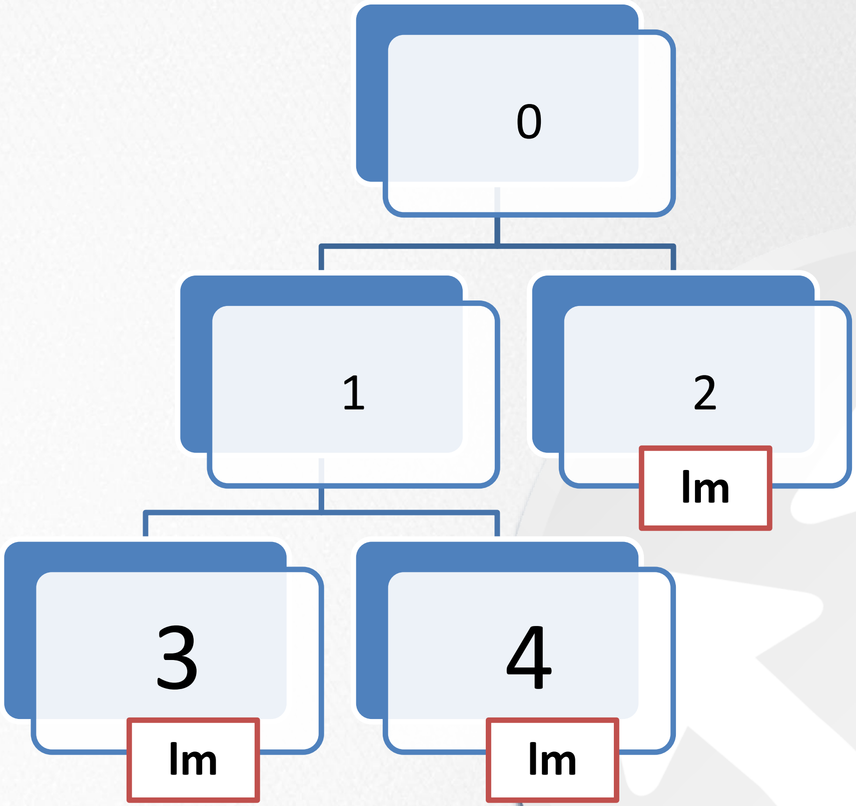
- **Greed is bad. Patience is good.**
  It is better to slowly approach your solution than arrive at an answer directly.
  - More accurate solutions

REVIEW

- M5
- Bagging
- Boosting
- Random Forest
- Simple Gradient Boosting
- Adaboost
- Tuning Parameters

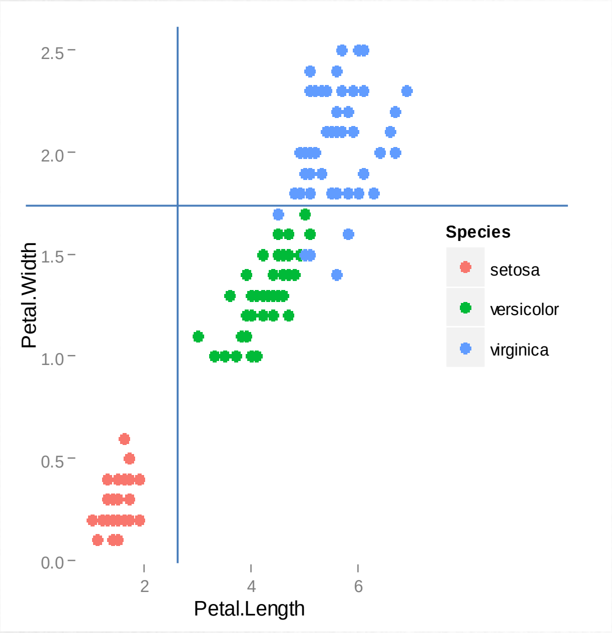
Tree Enhancement: M5

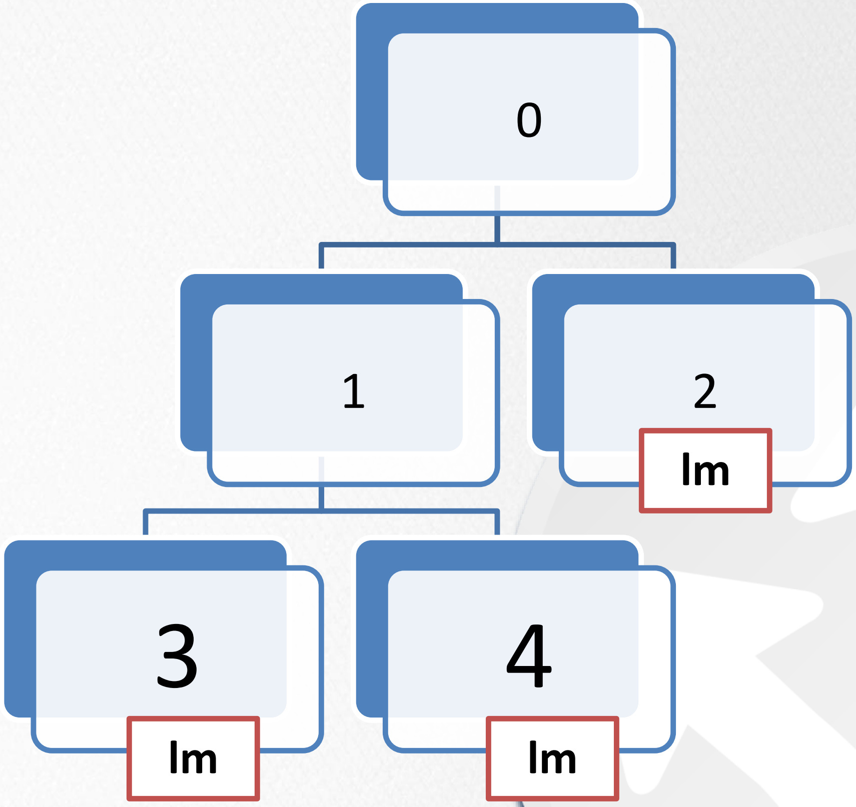
- **Wisdom of the Crowd!**
- Having one value represent the entirety of the node leaves information in the node.
- Function in the node is a simple average
- Use something better
  - **M5** put linear models in nodes of trees



M5 Tree Enhancement (cont.)

- **Greed is bad**
  - linear models are built on the residuals of the tree model.
  - Models are recursive





Bagging Models

- Brieman:
  "Bagging is a general approach that uses bootstrapping in conjunction with any regression (or classification) model to construct an ensemble."

```
for i = 1 to m do
    Generate a bootstrap sample of the original data
    Train an unpruned tree model on this sample
end
```

$$\hat{y} = \frac{\sum_i \hat{y}_i}{m}$$

Bagging Notes

- Lowers variance
  - Increases stability
  - Has less effect on lower variance models (e.g. linear models)
  - More effect on weak learners

- Disadvantages
  - Computational cost → but parallelizable
  - Reduces Interpretability

Random Forest

- **Wisdom of the Crowds:** Bagging
- **Greed is bad: consider subset of predictors at each split**

Select the number of models to build, m
for *i = 1 to m* **do**
    Generate a bootstrap sample of the original data
    Train a tree model on this sample
    for *each split* **do**
        Randomly select k $(< P)$ of the original predictors
        Select the best predictor among the k predictors and partition the data
    end
    Use typical tree model stopping criteria to determine when a tree is complete (but do not prune)
end

TUNING PARAMETER

m_{try} : number of predictors to use at each split

- **regression** 1/3rd of number predictors
- **classification** sqrt(number of predictors)

➲ Kuhn: "Starting with five values of k that are somewhat evenly spaced across the range from 2 to P".

Advantages

- No overfitting
- More trees better (limited by computation time/power only)
- In caret, parameters are considered independently
- Because each learner is selected independently of all previous learners, Random Forests is robust to a noisy response
- Computationally efficient -- each tree built on subset of predictors at each split.
- Use any tree variants as "base learner": CART, ctree, etc

Boosting

- Single models work;
  - Multiple models work better

- Idea is simple:
  - **Fit first** model: $\hat{y}_1 \sim f_1\,(x)$
  - **Fit** errors/residuals: $\hat{y}_2 = f_2(y - \hat{y}_1) = f_2\big(y - f_1(x)\big) = f_2\,(x)$
  - **Iterate:** $\hat{y}_i = (y - \hat{y}_{i-1}) \sim f_i\,(x)$
  - **Predict**: $\hat{y} \sim \sum_i f_i\,(x)$

Boosting Notes

- Additive models
- Works best with "weak learners"
  - i.e. ungreedy, low bias, low variance
  - ~~Any~~ Most models with a tuning parameter can be a weak learner
  - Trees are excellent weak learners
    - Weak → "restricted depth"
- Residuals or errors define a gradient
- Interpreted as forward step-wise regression with exponential loss

REVIEW: BOOSTING

- Patience
  - Iterative, repeatedly model residuals
  - Ensemble technique (?)
- Powerful, tends to over-fit
  - Early stopping
  - Learning rate
  - Gradient boosting / stochastic gradient boosting / Gradient boosting machines

Caret: method="gbm"

- Applied to any base learner, commonly trees
- Similar results to bagging
- Computational more expensive

Simple Gradient Boosting

Select tree depth, D, and number of iterations, K

Compute the average response, $\overline{y}$, and use this as the initial predicted value for each sample

for k = *1 to* K **do**

> Compute the residual, the difference between the observed value and the *current* predicted value, for each sample
>
> Fit a regression tree of depth, D, using the residuals as the response
>
> Predict each sample using the regression tree fit in the previous step
>
> Update the predicted value of each sample by adding the previous iteration's predicted value to the predicted value generated in the previous step

end

Stochastic Gradient Boosting

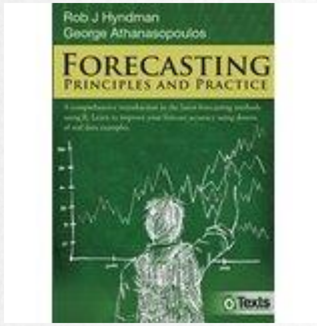
- Gradient Boosting Susceptible to Overfitting
  - Apply "regularization/shrinkage"
    - Use λ ("Learning Rate")
      Rather than add the entirety of the residuals, add a fraction of the residuals at each iteration.

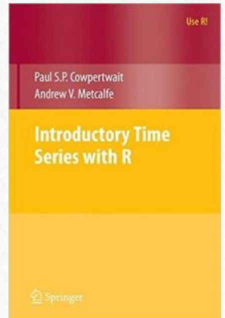
$$\hat{y} \sim \lambda \sum_i f_i\,(x) \qquad 0 < \lambda \leq 1$$

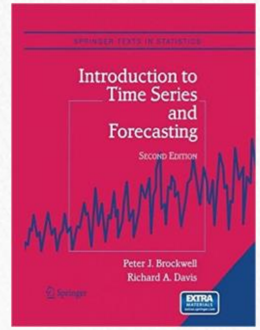
    - Small values for λ (~0.01) work best
    - λ ~ 1/computational time ~ 1/storage size
- Use bagging, as well
  - Bagging Fraction: a sample of data in each loop iteration

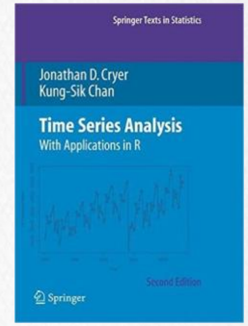
TIME SERIES ANALYSIS

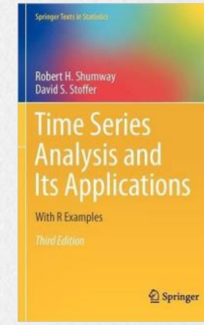
Resources











	Forecasting Principles and Practice	**Introductory Time Series with R**	**Introduction to Time Series and Forecasting**	**Time Series Analysis with Application in R**	**Time Series Analysis and Applications**
Author(s)	Hyndman* / Athanasopoulos	Copertwaite / Metcalfe	Brockwell / Davis	Cryer / Chan	Shumway / Stoffer
Level	Beginner	Beginner	Beginner	Intermediate	Intermediate / Advanced
Amazon	4.3 (11)	4.1 (26)	3.5 (21)	3.9 (5)	3.6 (5)
Notes	Free online				

*Also CRAN TASK VIEW: Time Series Analysis

Time Series Analysis (TSA) Goals

- Identify the nature of the phenomenon represented by the sequence of observations, and
- Forecasting (predicting future values of the time series variable).

Both of these goals require that the pattern of observed time series data is identified and more or less formally described … parametric

Time Series Analysis

- Observations are related through some index, commonly time
  - Software still requires tabular data
- There are several common questions we can ask regarding time …

Classic Forecast

- Value(s) of a *forecast variable* ("response") at certain given future time?
  - What are the values that affect the forecast.
  - What is the confidence of the forecast.

Examples:

- What will value of IBM stock be at Market Close Tomorrow
- What will be tomorrow's weather

Survival Analysis

- How much time will elapse until an event (e.g. “Death”)
  - What is the probability that a metric attains a value within a interval?
  - What factors affect this and by how much?
  - How does this compare to an alternative series?

Examples

- What is the *mean survival time* of someone afflicted with Ebola?
- How does this compare with patients who receive *VSV-EBOV* vaccine?

Poisson Processes

- How many events occur within a given (future) interval?

- Examples:
  - Customer Lifetime Value
  - Churn:

Change Detection

- Does the *response* or environment change over time
  - Is there structure to how things change over time
  - What is the structure to the response: *trend, seasonality*

- *Examples*
  - *Where are intrusions occurring in my network*
  - *Where are insurgents located*

Seems Like a Lot …

- Most (All?) of our ideas still apply

Tool Box

- R's Date/Time Classes
  - motley and varied
  - POSIXct (POSIXlt) supported by **Lubridate** and
- Base functions in stats package
- Forecast and Related Packages (Hyndman)

CLASSIC FORECASTS

Differentiate Models

- "Explanatory" Model

$$\hat{y} = f(x_1, x_2, \ldots, x_n)$$

- "Time Series" Model

$$y_t = f(y_{t-1}, y_{t-2}, \ldots, y_{t-n})$$

  - Naive model: $y_t = y_{t-1}$

- "Mixed" Model

$$y_t = f(y_{t-1}, x_1, x_2, \ldots, x_n)$$

Use Existing Tools

- In "theory", you can use existing tools to develop forecasting models though it may not be advised.
  - Predictors may be highly correlated and so method robust to correlated predictors will perform better.
  - Measure performance by modified Cross Validation

Success Factors

- how well we understand the factors that contribute to the response
- how much data are available
- whether the forecasts can affect the thing we are trying to forecast.

Rolling Forecasting Origin

- Select the observation at time k+i for the test set, and use the observations at times 1,2,…,k+i−1 to estimate the forecasting model. Compute the error on the forecast for time k+i.
- Repeat the above step for i=1,2,…,T−k where T is the total number of observations.
- Compute the forecast accuracy measures based on the errors obtained.

Decompositional Forecasting

- Component of Time Series
  - Trend
  - Seasonality
  - Cycle

Handled by the **Forecast** Package

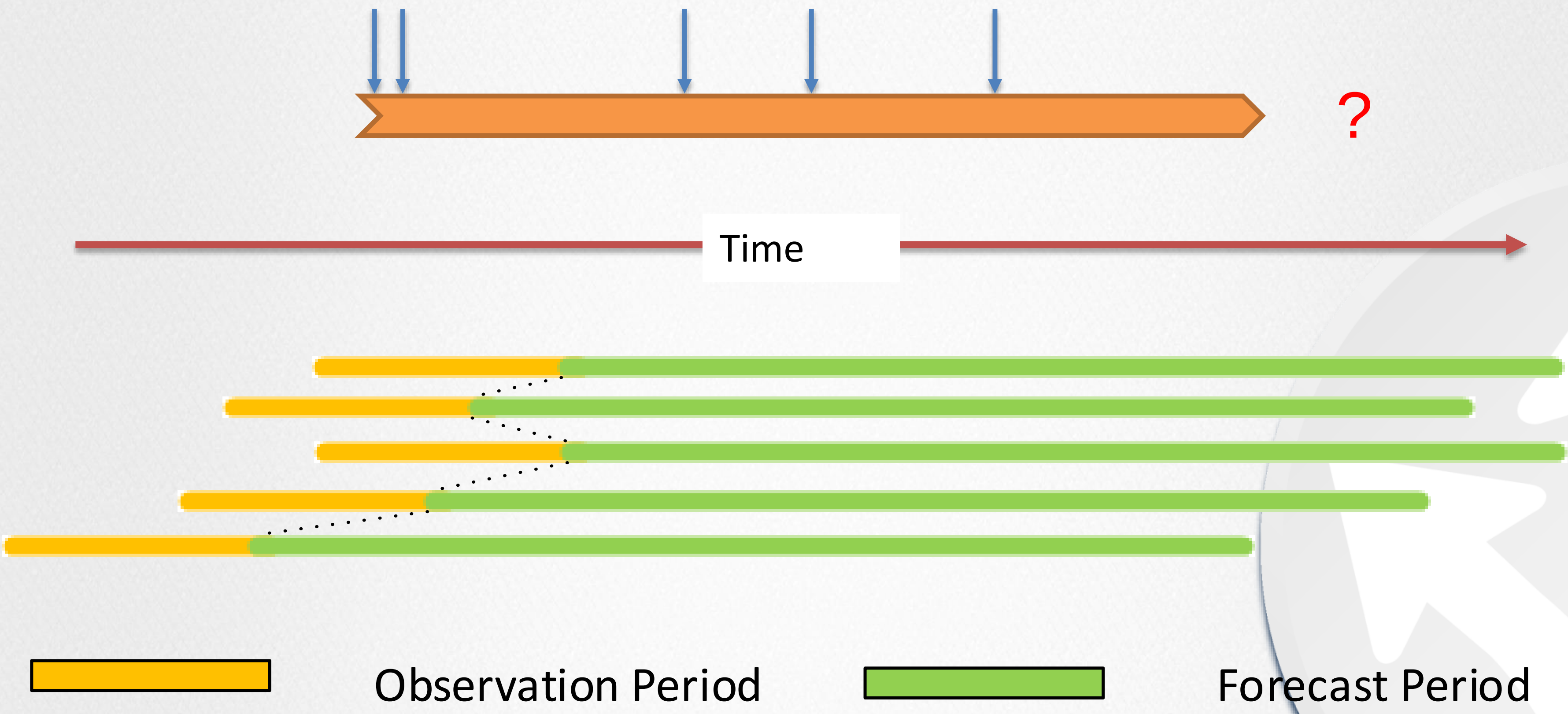
- Caveats:
  - Does not work well with TS without these characteristics.
  - Stationary Time Series Only

Time Series Ensembles

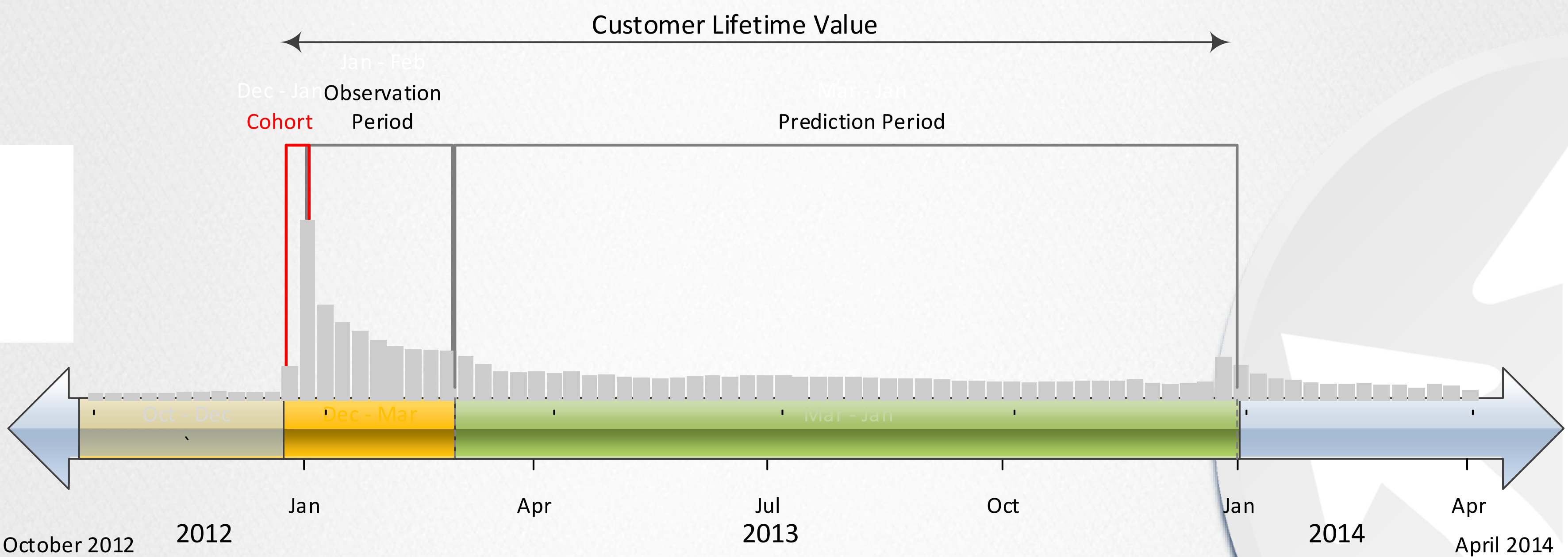
- Common to combine different types of models for a prediction:
  - Explanatory in addition to Decompositional
  - Can be done either sequentially or though averaging.

POISSON PROCESSES

Customer Timelines

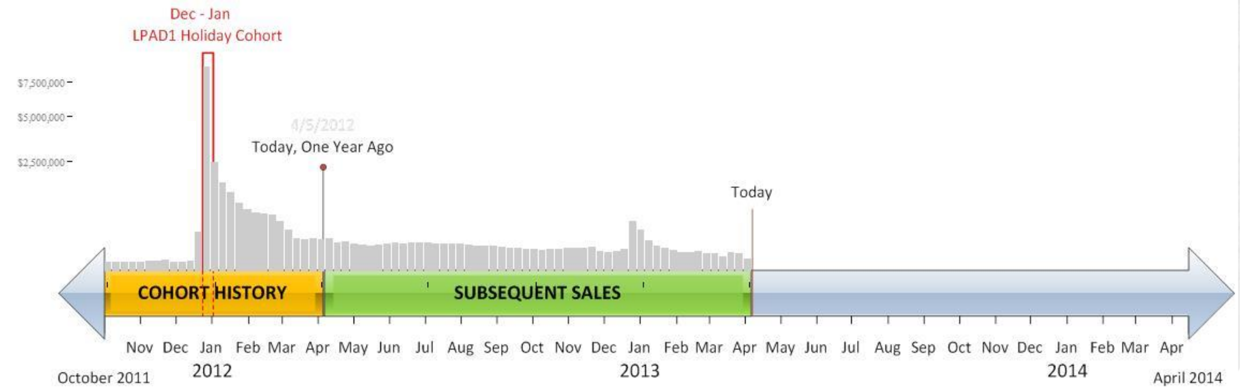


Modeling


Customer Lifetime Value
Cohort
Observation Period
Prediction Period
Jan
Apr
Jul
Oct
Jan
Apr
October 2012
2012
2013
2014
April 2014

CUSTOMER LIFETIME VALUE

1. **Use historical customer** segments to predict subsequent historical content purchase



2. **Apply this predictive:** behavior to current customers to predict future

3. **Proscribe**: Test hypotheses to increase content consumption behavior (and retrain/tweak model to reforecast future sales)