

- Building an A.I. Cloud  
What We Learned from PredictionIO



# Simon Chan

Sr. Director, Product Management, Salesforce

Co-founder, PredictionIO

PhD, University College London

[simon@salesforce.com](mailto:simon@salesforce.com)

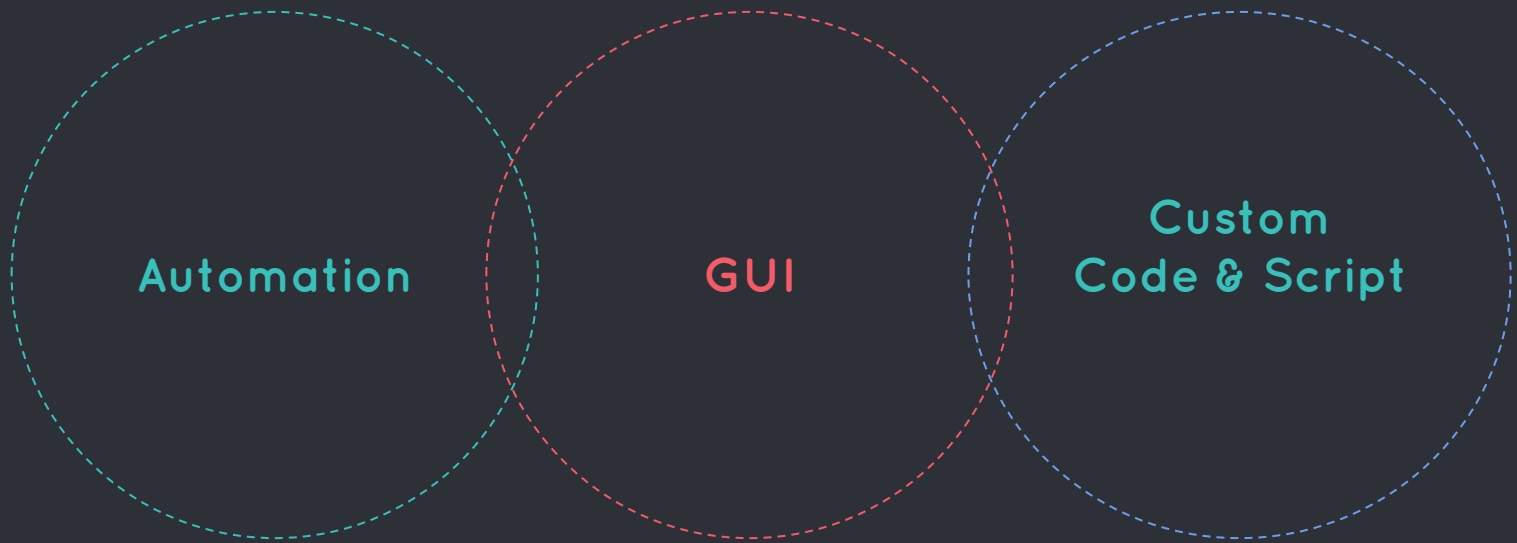
# The A.I. Developer Platform Dilemma



Simple ○————○ Flexible



- 3 Approaches to Customize Prediction



Simple ☐ ☐ Flexible



# 10 KEY STEPS

to build-your-own A.I.

P.S. Choices = Complexity

- One platform, build multiple apps. Here are 3 examples.

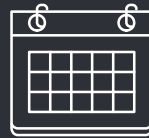
### 1. E-Commerce

Recommend  
products



### 2. Subscription

Predict churn



### 3. Social Network

Detect spam



## Let's Go Beyond Textbook Tutorial

**// Example from Spark ML website**

**import org.apache.spark.ml.classification.LogisticRegression**

*// Load training data*

**val** training = sqlCtx.read.format("libsvm").load("data/mllib/sample\_libsvm\_data.txt")

**val** lr = **new LogisticRegression**().setMaxIter(10).setRegParam(0.3).setElasticNetParam(0.8)

*// Fit the model*

**val** lrModel = lr.fit(training)



1

# Define the Prediction Problem

Be clear about the goal

## Define the Prediction Problem



### Basic Ideas:

- What is the Business Goal?
  - Better user experience
  - Maximize revenue
  - Automate a manual task
  - Forecast future events
- What's the input query?
- What's the prediction output?
- What is a good prediction?
- What is a bad prediction?



2

## Decide on the Presentation

It's (still) all about human perception

## Decide on the Presentation

Mailing List and Social Network, for example, may tolerate false predictions differently

		Actual	
		NOT SPAM	SPAM
Predicted	NOT SPAM	True Negative	False Negative
	SPAM	False Positive	True Positive

## Decide on the Presentation



## Some UX/UI Choices:

- Toleration to Bad Prediction?
- Suggestive or Decisive?
- Prediction Explainability?
- Intelligence Visualization?
- Human Interactive?
- Score; Ranking; Comparison; Charts; Groups
- Feedback Interface
  - Explicit or Implicit

3

## Import Free-form Data Source

Life is more complicated than MNIST and MovieLens datasets

## ● Import Free-form Data Sources



## ○ Some Types of Data:

- User Attributes
- Item (product/content) Attributes
- Activities / Events

Estimate (guess) what you need.

● Import Free-form Data Sources



## ○ Some Ways to Transfer Data:

- Transactional versus Batch
- Batch Frequency
- Full data or Changed Delta Only

Don't forget continuous data sanity checking

4

## Construct Features & Labels from Data

Make it algorithm-friendly!

## Construct Features & Labels from Data



## Some Ways of Transformation:

- Qualitative to Numerical
- Normalize and Weight
- Aggregate - Sum, Average?
- Time Range
- Missing Records

## Label-specific:

- Delayed Feedback
- Implicit Assumptions
- Reliability of Explicit Opinion

Different algorithms may need different things

## Construct Features & Labels from Data

Qualitative to Numerical

// Example from Spark ML website - TF-IDF

```
import org.apache.spark.ml.feature.{HashingTF, IDF, Tokenizer}

val sentenceData = sqlContext.createDataFrame(Seq(
  (0, "Hi I heard about Spark"),
  (0, "I wish Java could use case classes"),
  (1, "Logistic regression models are neat")
)).toDF("label", "sentence")

val tokenizer = new Tokenizer().setInputCol("sentence").setOutputCol("words")
val wordsData = tokenizer.transform(sentenceData)
val hashingTF = new HashingTF()
  .setInputCol("words").setOutputCol("rawFeatures").setNumFeatures(20)
val featurizedData = hashingTF.transform(wordsData)
val idf = new IDF().setInputCol("rawFeatures").setOutputCol("features")
val idfModel = idf.fit(featurizedData)
val rescaledData = idfModel.transform(featurizedData)
```

5

## Set Evaluation Metrics

Measure things that matter

## ● Set Evaluation Metrics



## ○ Some Challenges:

- How to Define an **Offline** Evaluation that Reflects Real **Business Goal**?
- Delayed Feedback (again)
- How to Present The Results to **Everyone**?
- How to Do Live A/B Test?

6

## Clarify “Real-time”

The same word can mean different things

- Clarify “Real-time”



## ○ Different Needs:

- Batch Model Update, Batch Queries
- Batch Model Update, Real-time Queries
- Real-time Model Update, Real-time Queries

When to Train/Re-train for Batch?

7

## Find the Right Model

The “cool” modeling part - algorithms and hyperparameters

## Find the Right Model

Example of model hyperparameter selection

**// Example from Spark ML website**

*// We use a ParamGridBuilder to construct a grid of parameters to search over.*

**val** paramGrid = **new ParamGridBuilder**()

.addGrid(hashingTF.numFeatures, **Array**(10, 100, 1000))

.addGrid(lr.regParam, **Array**(0.1, 0.01)).build()

*// Note that the evaluator here is a BinaryClassificationEvaluator and its default metric*

*// is areaUnderROC.*

**val** cv = **new CrossValidator**()

.setEstimator(pipeline).setEvaluator(**new BinaryClassificationEvaluator**)

.setEstimatorParamMaps(paramGrid)

.setNumFolds(2) *// Use 3+ in practice*

*// Run cross-validation, and choose the best set of parameters.*

**val** cvModel = cv.fit(training)

● Find the Right Model



## ○ Some Typical Challenges:

- Classification, Regression, Recommendation or Something Else?
- Overfitting / Underfitting
- Cold-Start (New Users/Items)
- Data Size
- Noise



8

## Serve Predictions

Time to Use the Result

## Serve Predictions



## Some Approaches:

- Real-time Scoring
- Batch Scoring

Real-time Business Logics/Filters is often added on top.

9

## Collect Feedback for Improvement

Machine Learning is all about “Learning”

## ● Collect Feedback for Improvement



## ○ Some Mechanism:

- Explicit User Feedback
  - Allow users to correct, or express opinion on, prediction manually
- Implicit Feedback
  - Learn from subsequence effects of the previous predictions
  - Compare predicted results with the actual reality

10

## Keep Monitoring & Be Crisis-Ready

Make sure things are *still* working

● Keep Monitoring & Be Crisis-Ready



## ○ Some Ideas:

- Real-time Alert (email, slack, pagerduty)
- Daily Reports
- Possibly integrate with existing monitoring tools
- Ready for production rollback
- Version control

For both **prediction accuracy** and **production issues**.

## ● Summary: Processes We Need to Simplify

- Define the Prediction Problem

- Decide on the Presentation

- Import Free-form Data Sources

- Construct Features & Labels from Data

- Set Evaluation Metrics

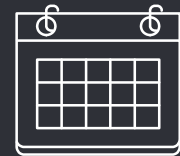
- Clarify “Real-Time”

- Find the Right Model

- Serve Predictions

- Collect Feedback for Improvement

- Keep Monitoring & Be Crisis-Ready



- The Future of A.I.



PredictionIO

is the automation of A.I.



Thanks! Any Questions?

**WE ARE HIRING.**

simon@salesforce.com

@simonchannet