13th - 17th May 2019 – Kuala Lumpur

Fraud Model Development and Deployment in SAS FM

Session 1: Introduction to Fraud Modeling in SAS FM

Introduction to Fraud Modeling in SAS FM

Just a Background Check

- Experience with
  - SAS FM
  - SAS Datastep programming
  - SAS Macros
  - Modeling
  - Fraud modeling
  - Viya and newer tools
  - Python and other languages

???

Introduction to Fraud Modeling in SAS FM

Objectives

- Get a deep insight into how the SAS FM analytic engine works
- Get an understanding of how fraud models are currently developed for SAS FM
- Get familiar with related SAS FM topics such as message layouts, consortium etc.

- This is not a modeling 101
- We will not get in-depth into the feature engineering aspect of fraud models

Final objective is for you to have the necessary information to successfully develop and deploy 'a' model in SAS FM

Introduction to Fraud Modeling in SAS FM

Topics

- Introduction to fraud modeling and challenges
- An overview of SAS FM models
- Overview of the modeling process
- A sampler of SAS FM models

Introduction to Fraud Modeling in SAS FM

Before we start, a list of terms / phrases that we will use and their meaning in SAS FM

Introduction to Fraud Modeling in SAS FM

Selected SAS FM Vocabulary

- Consortium
- Model
- Model performance
- API
- ODE
- Montetary / non-monetary transactions
- Cards / payments / deposits / checks / merchant

- Recommend keeping a copy of this slide by your side for the entirety of these presentations

Introduction to Fraud Modeling in SAS FM

Introduction

Introduction to Fraud Modeling in SAS FM

Introduction

- Fraud modeling for the finance industry is predominantly a supervised two-class classification problem
  - Objective of the model is to classify frauds vs. non-frauds
  - Other variants also exist: e.g. return vs. non-return
  - Unsupervised models used in limited circumstances

- Good fraud models are built using a combination of:
  - Well conditioned data
  - Good feature engineering
  - Application of <u>appropriate</u> data science and machine learning methodologies
  - Behavioral information from various channels will also immensely help

Introduction to Fraud Modeling in SAS FM

Introduction

- 96% of card issuers, 77% of banks that issue checks, 24% of banks that offer ACH money transfer, 13% of banks that offer wire transfers experienced fraud losses in the US in 2016 [1].

- However minimizing fraud losses is not always the primary goal for deploying fraud models
  - Reducing customer friction; i.e. false positives
  - Revenue maximization
  - Mitigating reputational risk

[1] 2017 Financial Institution Payments Fraud Mitigation Survey, Payments, Standards, and Outreach Group, Federal Reserve Bank of Minneapolis, January 2018

Introduction to Fraud Modeling in SAS FM

Nature of Fraud

- Fraud is complex
  - Needs models that can capture and understand complex behavior
  - Models need to be driven by Big Data

- Fraud requires real time decisions
  - Necessary to block fraud when it occurs, not after the fact in order to minimize losses

- Fraud adapts and evolves continuously
  - Fraudsters are sophisticated; fraud models need to keep up or get ahead
  - Fraudsters may just focus elsewhere for a brief period only to return again

Introduction to Fraud Modeling in SAS FM

Challenges in Fraud Modeling

- Extreme class imbalance
  - Between 5 bps (cards) and < 0.1 bps (commercial payments) of fraud
  - False positive suppression is the real challenge
- Model stability is a key concern
  - Given the class imbalance, models are built to be very sensitive to even the faintest of signals
  - Any little perturbation to the model inputs can have extreme effects; score distributions can go haywire

Introduction to Fraud Modeling in SAS FM

Challenges in Fraud Modeling

- Data quality issues
  - Mostly transactional information coming from the network sources; data quality is at the mercy of the feeding networks for the most part
    - Some networks are better than others (e.g. Visa, MC compared to regional ATM networks)
  - Amount of possible enrichment is limited due to SLA requirements

- Data availability issues
  - Many banks may not be able to / willing to share sufficient historical data

- Quality issues in fraud reporting
  - Missing fraud / first party fraud
  - Incomplete transaction level reporting even on true fraud cases

Introduction to Fraud Modeling in SAS FM

Challenges in Fraud Modeling

- Stringent SLA requirements due to real-time scoring
  - Most transactional system flows allocate < 100ms for the fraud system
  - Out of which 90% of the time is taken by other tasks (transaction enrichment, fetching profiles, rules, alert creation etc.)
- Governance
  - Finance industry is very conservative in using black-box models due to the complex legal landscape
- Misplaced expectations and conceptions
  - Huge misconceptions about ML and AI creates wrong expectations

Introduction to Fraud Modeling in SAS FM

Models in SAS FM

Introduction to Fraud Modeling in SAS FM

Models in SAS FM

- Models in SAS FM are / should be built to have very high analytic and technical performance
- Threshold for ‘high’ analytic performance can be fuzzy; it depends on the channel, customer, region etc.
  - Nevertheless the reputation of SAS FM has been built on analytically high performing models
- Technical performance is almost always precisely specified in terms of SLAs
  - Models are almost always built to have a execution time in the order of single digit milliseconds

Introduction to Fraud Modeling in SAS FM

Models in SAS FM

- Models in SAS FM are constructed as "packages" which are swappable modules within the ODE
- The term 'model' typically refers to the model package
  - When disambiguation is required, it will be explicitly stated so.
- A model package contains the following:
  - A compiled SAS macro catalog or collection of files that contain(s):
    - Module for resolving the signature keys (mandatory depending on nature of key)
    - Modules for additional pre-processing steps (optional)
    - Module for updating the signatures (optional)
    - Module for transaction scoring (mandatory)
    - Signature definitions (mandatory)
  - Lookup datasets (loaded as hashes and used to lookup data within the model)
- A typical scoring module will contain multiple models due to segmentation

Introduction to Fraud Modeling in SAS FM

A Little Divergence: Very Quick Introduction to Signatures

Introduction to Fraud Modeling in SAS FM

Very Quick Introduction to Signatures

- Signatures are the mechanism through which behavioral information about entities is maintained in SAS FM.
  - Can be fetched and updated in real-time on every transaction
- Are essentially a collection of arrays and scalars associated with a given entity
  - As long as a data structure can be decomposed into individual numeric or character arrays or scalars, it can be part of the signature.
  - With a bit of creativity, various complex data structures (FILO, stacks, graphs etc.) can be constructed

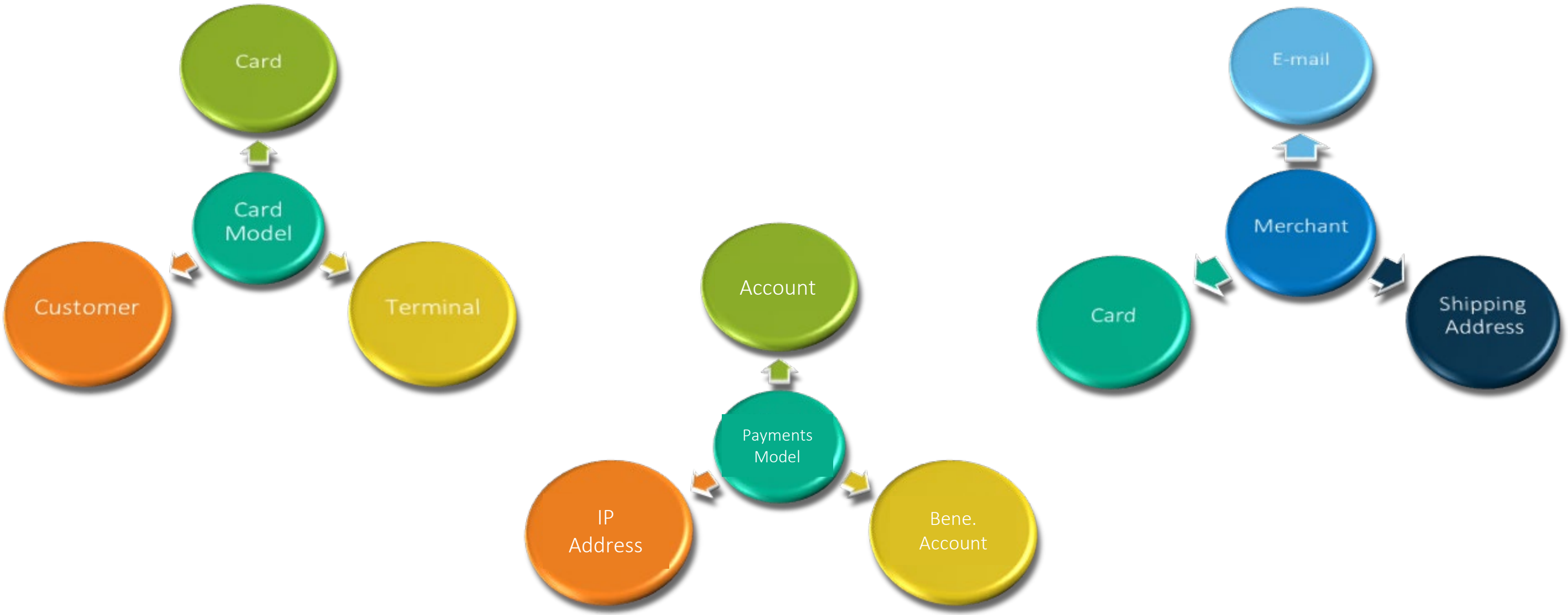
Introduction to Fraud Modeling in SAS FM

Very Quick Introduction to Signatures

- A signature segment is assigned to a given entity type.
  - E.g. card numbers, user IDs, beneficiary accounts, user IDs etc.
  - A signature segment is commonly referred to as a Z segment
- Each instance of a signature segment is keyed on a particular entity value
  - E.g. card signatures are keyed on card numbers
  - Keys can be fixed (e.g. card numbers) or ‘resolved’ (e.g. beneficiary account number by concatenating the account number with the bank ID)
- Technically you can define up to 16 signature segments (Z00 – Z15)
  - Typical model uses less than 4 segments
- The choice of signatures and its contents are one of the key design considerations during model development

Introduction to Fraud Modeling in SAS FM

Very Quick Introduction to Signatures



Introduction to Fraud Modeling in SAS FM

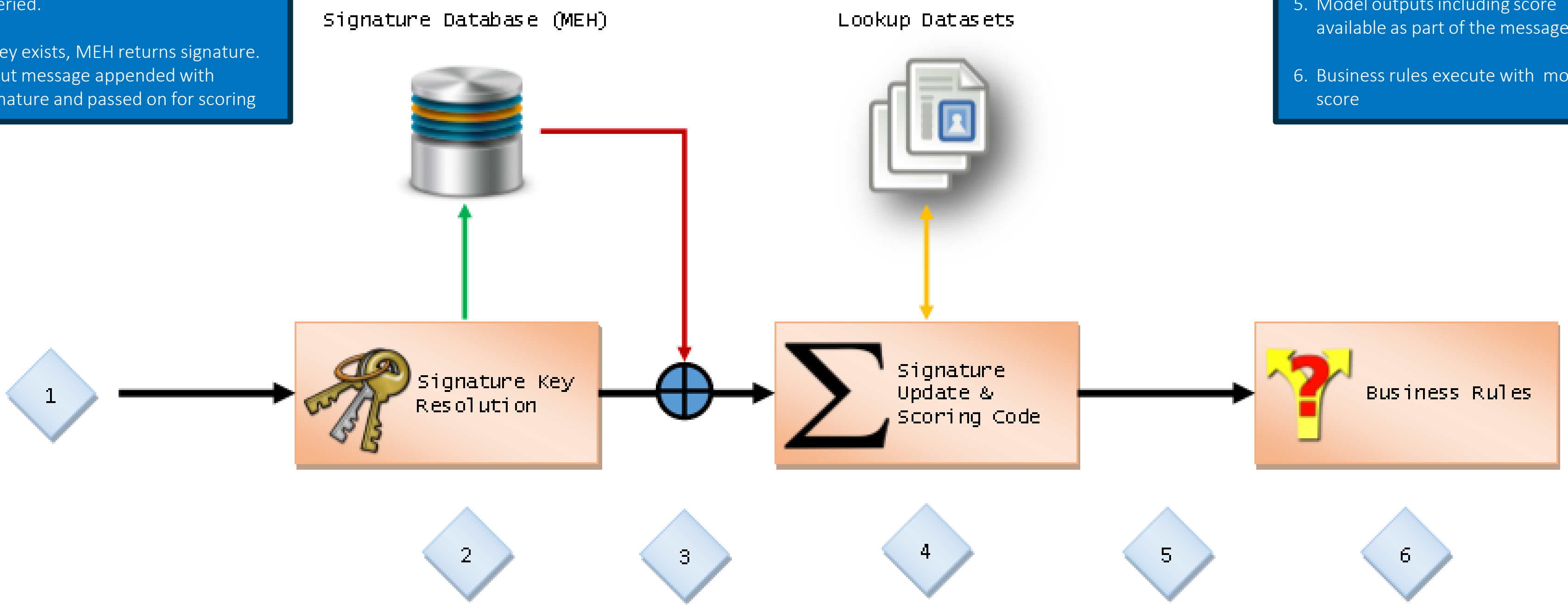
Now back to Models in SAS FM

Introduction to Fraud Modeling in SAS FM

High Level Process Flow

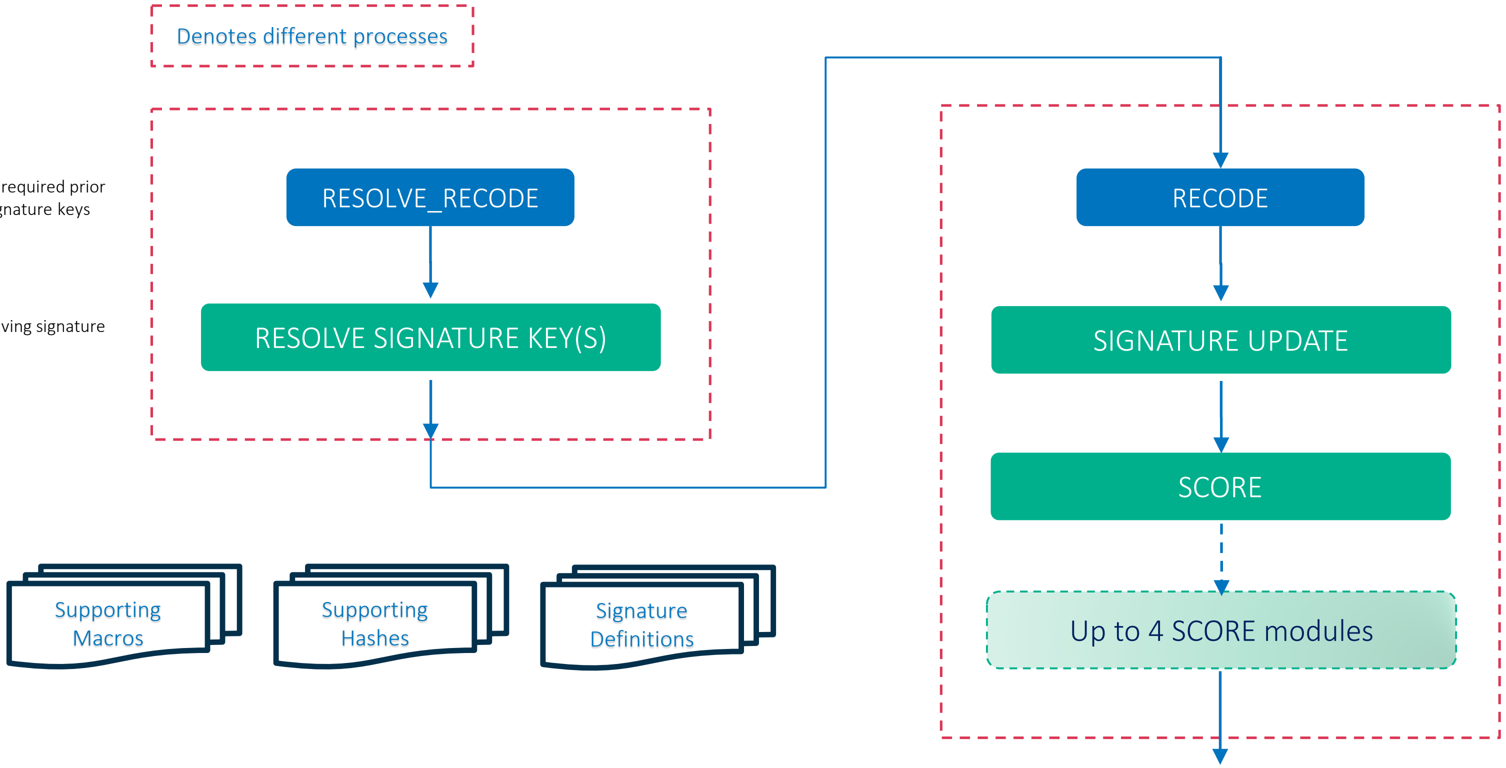
1. Input message arrives from ODE
2. Signature keys resolved and MEH queried.
3. If key exists, MEH returns signature. Input message appended with signature and passed on for scoring

4. Signatures updated based on current transaction and score produced
5. Model outputs including score available as part of the message
6. Business rules execute with model score



Introduction to Fraud Modeling in SAS FM

Flow Between Model Package Components



Preprocessing required prior to resolving signature keys

Logic for resolving signature lookup keys

Preprocessing prior to signature updates and scoring

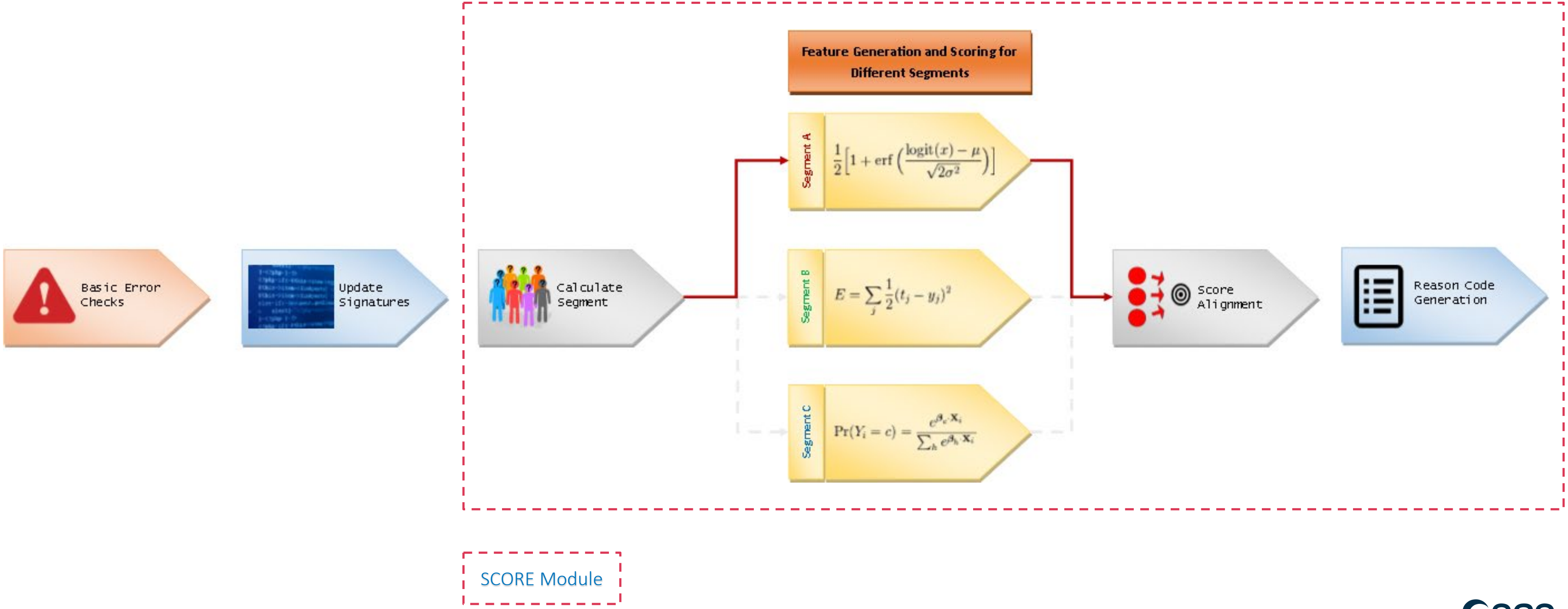
Code for updating signatures

Actual model code including segmentation, feature generation and scoring

Up to 4 models can be stacked to sequentially execute on a single transaction

Introduction to Fraud Modeling in SAS FM

Flow within Signature Update and Scoring Codes



Introduction to Fraud Modeling in SAS FM

Code Structure

- All model code runs within a SAS datastep in SAS FM
- Therefore all model code should be written to run inside a SAS datastep
  - PROCs cannot be called inside any of the model code
  - Can use SAS macros (heavily used)

```
data _null_;
   /* Read transaction from an incoming message stream */
   input stdin lrecl = &MAXLEN;
   %resolve_recode;
   %resolve_signature_keys;
run;

data _null_;
   /* Read transaction from an incoming message stream */
   input stdin lrecl = &MAXLEN;
   %recode;
   %update_signatures;
   %score1;
   %score2;
run;
```

Introduction to Fraud Modeling in SAS FM

Python Models

- Heavily utilizes the OOP nature of Python programming
- Signatures and models are defined via their own classes
  - Other modules are implemented in the form of methods which are invoked by the engine
  - Invocation order is similar to SAS data-step model flow

```python
import numpy as np

class CARD_SIG0100:

    def __init__(self):
        self.reset()
        return

    # Place code to reset signature contents here
    def reset(self):
        self.seconds = [None] * 10
        return

    # Place code to resolve signature key here
    def resolve(self, trx):
        key = trx.hqo_card_num
        return key

    # Place code to update signature on every transaction
    def update(self, trx):
        return;
```

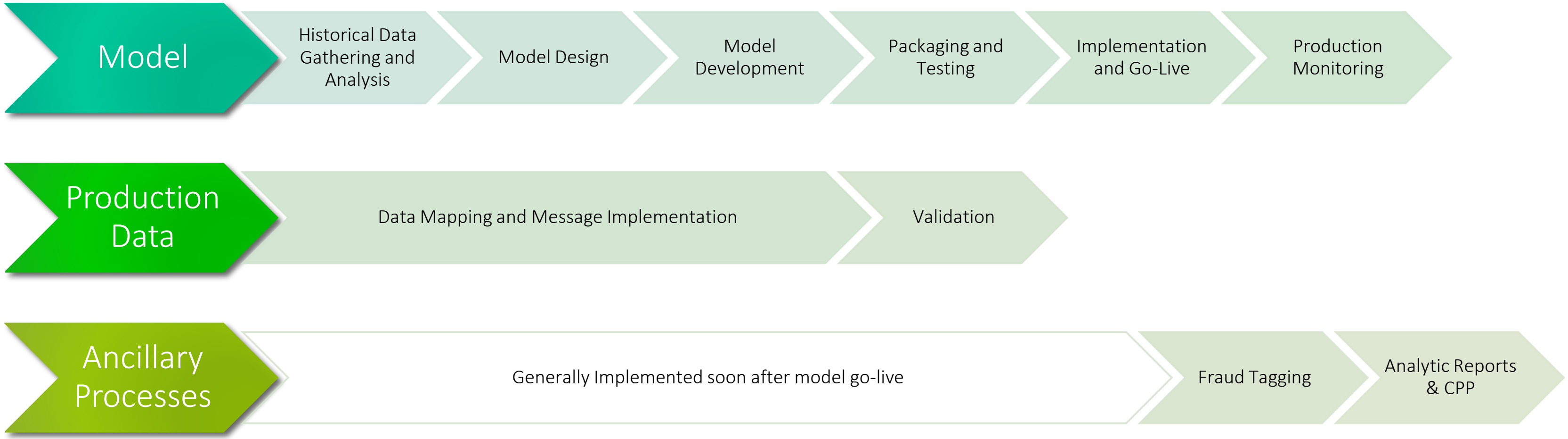
```python
from ose import ModelResult
import math as m

# trx is the Transaction object
# Individual fields are referenced as trx.<field_name>
# E.g. trx.hqo_card_num

class test_model:

    # One-time initialization code here.
    def __init__(self):
        return

    # Place code to score a transaction here.
    # This method should return a ModelResult Object
    def score(self, trx):
        score = 1
        return ModelResult(self, score)

    def custom_function(self, parm):
        return
```

Introduction to Fraud Modeling in SAS FM

Model Development Process

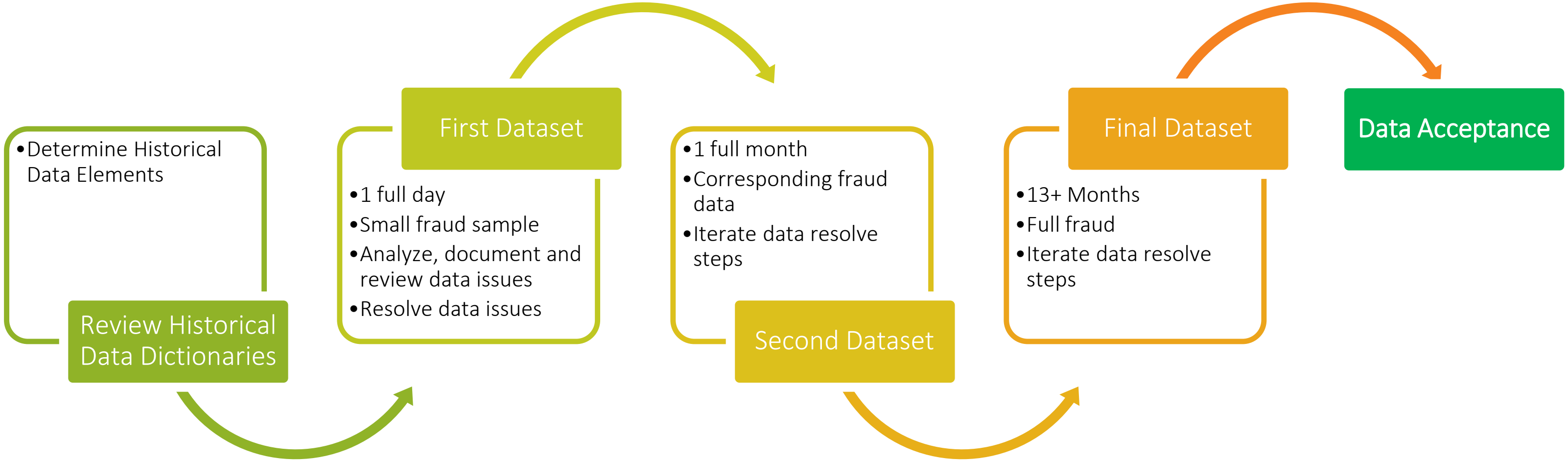
Introduction to Fraud Modeling in SAS FM

Modeling Process: First Time Models



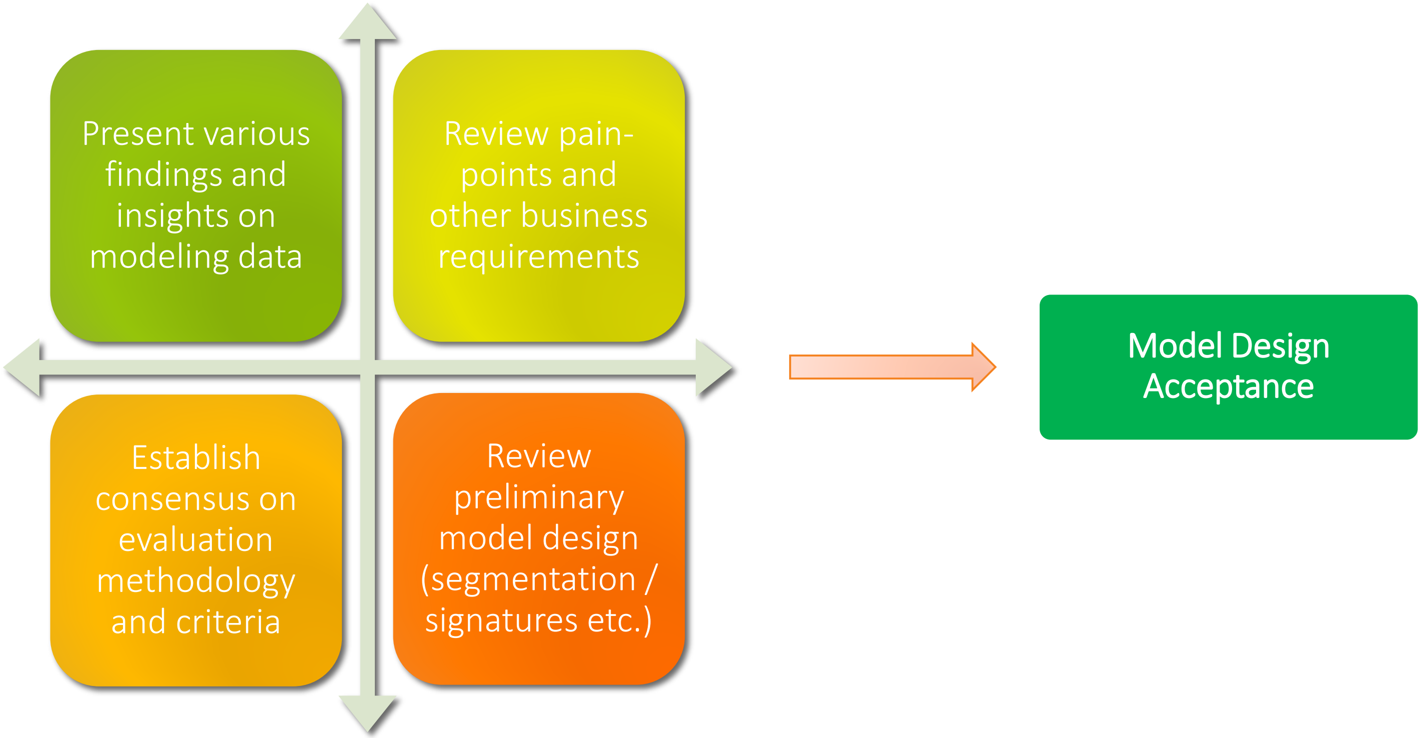
Introduction to Fraud Modeling in SAS FM

Historical Data Processing



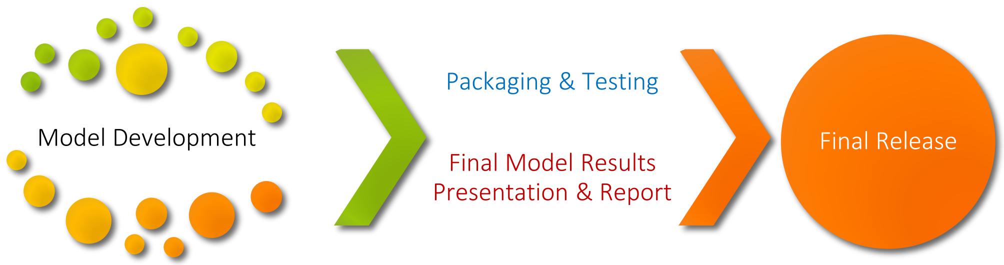
Introduction to Fraud Modeling in SAS FM

Model Design Meeting



Introduction to Fraud Modeling in SAS FM

Model Development and Delivery



- Some closed door initial preparation steps
- Iterations (Agile)

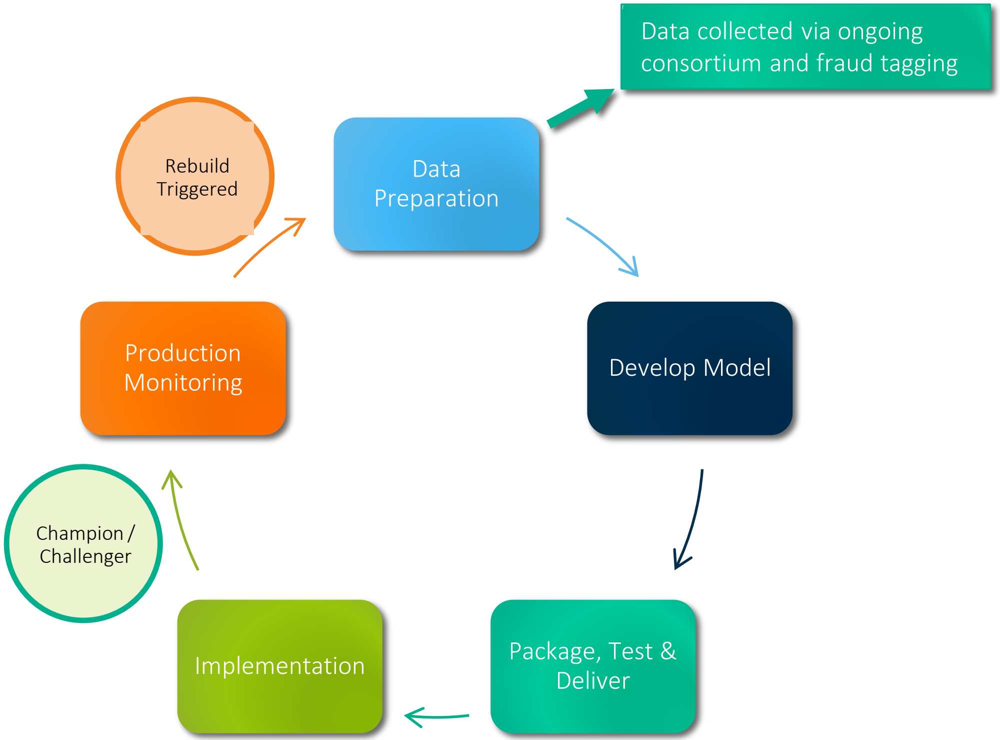
Introduction to Fraud Modeling in SAS FM

Key Milestones

- Data Acceptance
- Model Design Acceptance
- Final Model Report and Signoff
- Production Go-Live
- Establish Consortium Feeds
- Implement Ancillary Processes Including Fraud Tagging

Introduction to Fraud Modeling in SAS FM

Ongoing Process



Introduction to Fraud Modeling in SAS FM

Examples of Models Deployed in SAS FM

Introduction to Fraud Modeling in SAS FM

Bank of America Debit Card Model

- 25M transactions per day; peak of nearly 30M transactions per day
- Various relatively new modes of transactions:
  - Large and growing % of chip – on –chip transactions
  - Mobile wallet transactions
  - One-time use cards (barcodes)
- Based on a changed US fraud landscape :
  - 60% fraud is CNP
  - CP fraud heavily concentrated in non-chip enabled terminals such as AFDs and smaller merchants who have not migrated to chip enabled terminals
  - Growth of first party fraud (non-separable from true fraud due to lack of labeling)
  - Application fraud – related to identity theft (account in a state of fraud from day 1)
- Transaction level fraud rate < 0.05%

Introduction to Fraud Modeling in SAS FM

HSBC UK – Combined Credit, Debit and ATM Card Model

- Combination of debit and credit portfolios
- Consists of several large sub-portfolios including M&S.
- Fairly stable fraud and technology landscape :
  - Mature Chip deployment
  - Sizable, but still growing contact less payments
  - Fraud heavily leans toward CNP
  - Growing application fraud – related to identity theft (account in a state of fraud from day 1)
- Relatively very low rate of fraud for a card portfolio: ~ 0.02%
- Common customer ID that can link various accounts and cards to a single customer

Payment Fraud Model

NAB – Consumer and Commercial Payments Model

- Outbound and inbound payments
  - Instant and scheduled payments
  - Additional non-monetary information including logons, password changes, limit changes, session information etc.
- Multiple channels: online banking, mobile, phone, NABConnect (channel for commercial payments responsible for 25% of traffic)
- Personal banking customers average about 5 outbound payments per month; commercial customers average about 80.
- Fraud events range from about 200 – 600 per month; fraud rate ~ 0.001% (as a % of financial transactions)
  - ~ 0 fraud in commercial payments

Check Fraud Model

Bank of America – Check Fraud Model

- Incoming checks issued by bank account holders
  - Mix of personal (40%) and business accounts (60%)
- Various channels including clearing systems, ATM, mobile, branch, etc.
- Very limited information for model building and scoring
  - For e.g. cannot identify depositor (pay-to account) – not possible to track relationships
  - Even true date and time of check deposit not available when coming in from clearing systems
- Fraud rate ~ 0.007%

ACH Fraud Model

Bank of America – ACH Fraud Model

- Unsupervised model due to rarity of fraud events in historical (and production) data
  - Autoregressive neural network
- Data sources include ACH transactions (NACHA), user master files and session information
- Produces a batch score (ACH batch level score) and item level score
  - Alerts are created based on rules applied at the batch level, incorporating the batch score with other possible characteristics of batches
  - Once a batch is flagged for review, the entry score is used as a guide for deciding which entries to investigate first.