Risk Management
Model Management
Model Governance

MATLAB Computational Finance Conference
Paul Peeling, MathWorks
Agenda

- Guidance from the Regulators
- Realising Model Risk Management with MATLAB
  - Model Inventory
  - Model Development
  - Model Documentation and Review
  - Model Monitoring
- Interpretability of Machine Learning Models
Model Risk Management with MATLAB
ECB guide to internal models
General topics chapter
Mitigating Model Risk (ECB guide to internal models)

- Unified inventory ("registry") of models
- Consistency of modelling approaches
- Documentation standards – such that a 3rd party can implement
- Usage of models monitored on an ongoing basis
Model Risk Management Principles (SS 3/18)

1. Banks have an established definition of a model and maintain a **model inventory**
2. Banks have implemented an effective **governance framework**, policies, procedures and controls to manage their model risk.
3. Banks have implemented a robust **model development and implementation** process, and ensure appropriate use of models.
4. Banks undertake appropriate **model validation and independent review** activities to ensure sound model performance and greater understanding of model uncertainties.
Model Risk Management Frameworks

Concluding remarks

An effective model risk management framework is … an integrated and iterative process supported by a strong governance culture.
Reality and Vision

FROM THIS

Data Madness

TO THIS

Poor Quality Models
Regulatory Scrutiny
High Cost
Inconsistency
Frustrated Users

SDLC Vendor Driven

Expensive Vendors
Black-box technology

Reduced Cycle Time
Access to Tooling
Freedom to Analyse
Consistency
Lineage & Tractability
Low Cost

Agile/DevOps Focused

Data Federated at the Centre
Model Objects

Discovery
GRA as Service
getData
Model Execution
External Cloud with access to tooling

FROM THIS TO THIS

Data Madness

External Cloud
with access to tooling

Expensive
Vendors

Model
Objects

Black-box technology

API
API

Lineage & Tractability
Low Cost
Model Lineage throughout the Model Risk Management cycle

Risk management
Board and stakeholders
Regulator
Model Owners

IT, front-office
End-users

Model developers, quants, analysts.
Business lines

Independent model review and audit
Regulator

Definition and Development

Review and Approval

Monitoring, Reporting and Performance Assessment

Quality Assurance, Pre-implementation validation

Implementation and Deployment

Model Inventory

Model Validation, IT
Opportunity for Cost Savings with Model Risk Management

- Banks have 1,000s of models used in decision making
- One FTE can manage approximately 10 models
- One model per month can be validated
- Number of models increasing by 10-25% annually
- Model risk management can reduce costs by 30%

- 20% of institutions have fully adopted model risk management

Source: McKinsey 2017
Model Inventory

- Manage model validation projects
- Workflows and approval
- Action and report on model issues
- History and lineage of models
Data Governance

Sourcing data from multiple platforms and processes

Vetting data quality

No unified data model or interpretation

Inconsistent handling of data by location and over time

Historical data cannot be reproduced
Model Development Environment (MDE)

- Build models in steps
- Document as-you-go
- Trusted and reproducible
- Credit and Market Risk templates
- Explore, research and experiment

Definition and Development

- Build models in steps
- Document as-you-go
- Trusted and reproducible
- Credit and Market Risk templates
- Explore, research and experiment
Model Development Environment (MDE)

Mission: *Improve the pace, transparency and reproducibility of the model development and review processes through user-friendly tools that encourage a consistent approach.*

What are the pain points?

- Pace of building and reviewing models
- Ability to reproduce results
- Consistency of modeling approaches

What is the solution?

- MATLAB toolbox for risk modelling at HSBC
- Functions, apps, demos, and documentation
- Supports all stages of the workflow
- Leverages MATLAB toolboxes
- Target users: risk modellers and analysts
- Aims: improve pace, transparency, accuracy, reproducibility, consistency
Building Models as a Sequence of Steps

1. Data loading and pre-processing
2. Exploratory Data Analysis
3. Sampling and Segmentation
4. Feature Engineering
5. Train Models
6. Model Validation
7. Documentation
8. Deployment
Building a credit scorecard

The Model Development Environment includes interactive templates to rapidly develop risk models.
Model Review Environment (MRE)

- Validate models in any language
- What-if analysis
- Explain results in prose
- Benchmark and back-test
- Collaborate with developers
- Regulator reports
Collaborative Development and Review Process

- **Data**
  - Pre-processing
  - Exploratory Data Analysis
    - Data Review
    - Exploratory Data Analysis
    - Data Review
  - Feature Engineering
    - Data Review
  - Feature Selection
    - Candidate Models
      - Performance Assessment
      - Segmentation
        - Segmentation Analysis
      - Model Validation
        - Calibration
        - Deployment
  - Model Selection

Development activity

Review activity
Regulatory Documentation Authoring

Automatic generation of supporting analysis relieves the burden on model development and validation teams, by:

▪ Keeping visualizations and tables in sync with model developments (no copy-and-paste)
▪ Adhering to corporate styles, templates and quality output

A model development document typically ranges between 200 and 500 pages, and consumes 30% of the effort.

Our approach allows developers and reviews to focus effort on insight, assumptions and limitations.

Example: IRB Application Modules
1. Scoping
2. Technical model reviews
3. IT and Data
4. Use test and experience test
5. Permanent partial use and roll out plans
6. Financial reporting and stress testing
7. Internal audit and independent validation
8. Governance

Authoring of highlighted modules are supported by the Model Review environment.
Documentation Authoring Workflow
Model Execution Environment (MEE)

- Secure and controlled
- Scales horizontally and vertically
- Immediate deployment
- Hosted in-house or on-cloud
- Audit trail of model usage
Packaging, Production Deployment and Monitoring of Models

- Automated deployment of models into production without translation
- Integrate with existing front-end and back-end tech, or self-service platforms
- Performance of models monitored for operational and regulatory requirements
Model Monitoring Dashboard (MMD)

- Visuals and metrics
- Multiple views
- Real-time monitoring
- Configurable alerts
- Configurable layout
Model Monitoring Workflow

MDE

Publish

Analyse

Deploy

Monitor

MMD

MEE
Model Monitoring

![Model Health Dashboard](image)
Opportunity for Artificial Intelligence in Model Risk Management

Data-driven, agile

Interpretable, explainable

Pillar 1 Capital Models
Credit scoring

Anti money laundering
Customer churn
Explaining Machine Learning

- **Explaination Methods**
  - Model-specific
    - Logistic Regression
    - Decision Tree
    - xNN
  - Model-agnostic
    - Global
      - Predictor Importance
      - Partial Dependency
    - Local
      - LIME
      - SHAPley Values
Explaining Machine Learning

Partial Dependency Plots
Marginal effect of a feature on the prediction

SHAPley Values
How far is the instance away from the mean prediction?

Total

mean PD

PD of instance
Machine Learning Ease-of-Use

- Classification Learner app

- Protect Against Overfitting

- Learn About Model Types

- Compare Models with a Variety of Evaluation Metrics
Reducing the approval, deployment and refresh timeline
Implementation Options for Existing Models

- **Co-execution**
  - Directly execute Python and R models in MATLAB
  - Supported for validation, execution and monitoring

- **Transcription to MATLAB**
  - Automated for formats such as PMML
  - Guidance for SAS models
    - Videos
    - Cheat-sheets
    - 1-1 sessions with MathWorks
Implementation Challenges and Data Considerations

- Best-in-class tools embrace an Agile/DevOps approach
  - Version and configuration control is mandatory for traceability
  - Reviews, workflow, project management for complex software

- Support innovation in modelling
  - Reproducibility and performance across different platforms
  - Permit scrutiny and independent implementation
  - Reusing innovations in modelling and methodology in different tools

- Data considerations
  - Cleansing not always possible in source systems
  - Data modelling is not independent of risk modelling