

Single Day Event

AI/ML in Clinical Data Science: Challenges and best practices

Soma Bhadra,
Eli Lilly



Disclaimer

The views and opinions expressed in this presentation are those of speakers and do not reflect opinions of any other person or organization



Agenda

- Operational and Ethical Challenges
- Core Considerations for AI Implementation
- Implementation Approach
- Business Use Case



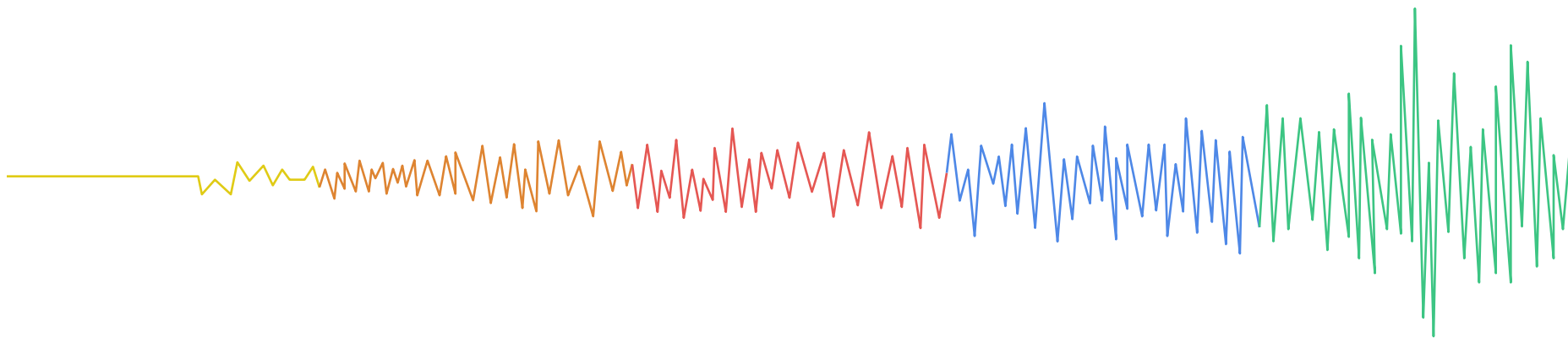
Challenges in AI implementation

AI challenges range from internal to external focus.

Internal



External



Data Quality

To maintain data consistency during development and implementation

Stakeholder Alignment

To balanced diverse Stakeholders expectations

Transparency

To clarify complex decisions process

Regulatory and Legal compliance

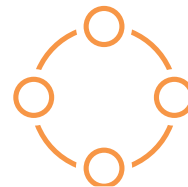
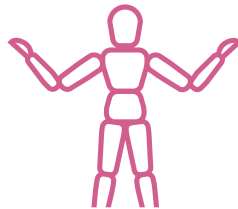
To safeguard evolving organizational systems

Reliability

To maintain consistent operational reliability



Implementation Approach



Prioritization & Alignment

Align every AI initiative with business strategy and compliance

Data Readiness/Governance

Build a robust, compliant data foundation to ensure trust, quality, and reusability across initiatives

Model Development

Develop transparent, validated models that balance innovation with scientific and regulatory rigor

Scaled Deployment and Integration

Integrate AI solutions seamlessly into clinical workflows to enable scalability and measurable value realization.

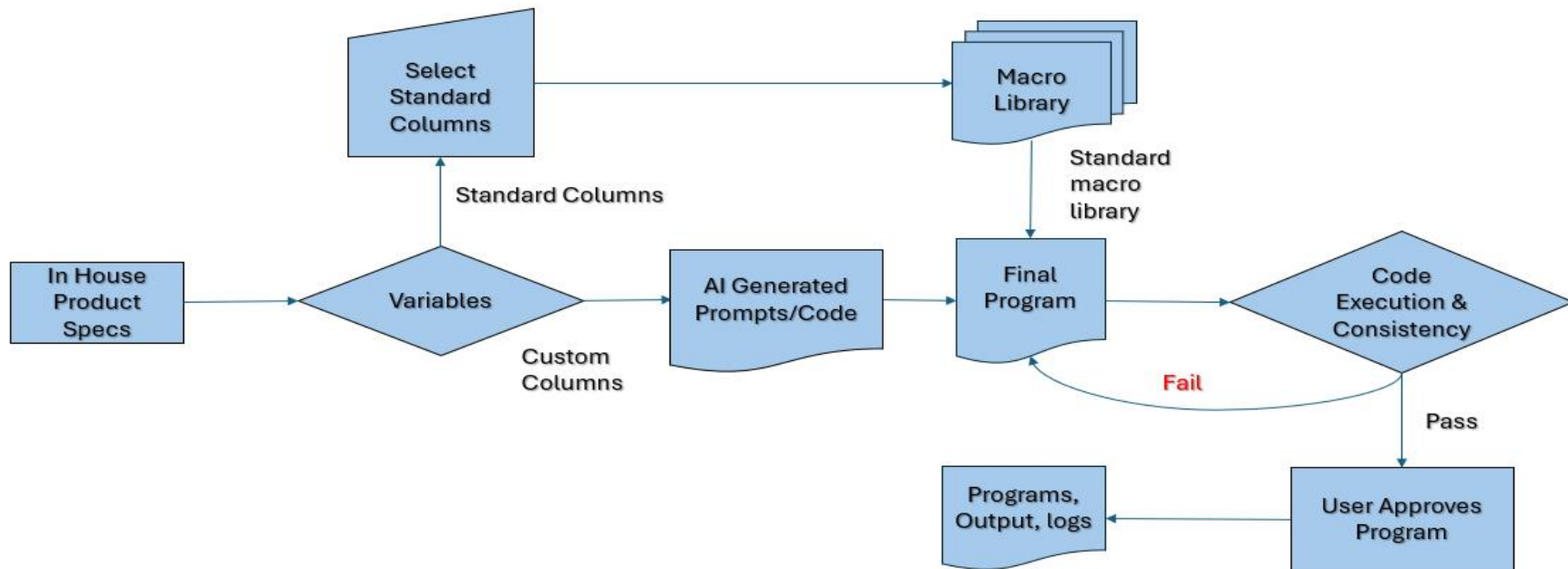
Acceptance, Adoption, and OCM

Foster enterprise-wide adoption through strong leadership sponsorship, change enablement, and capability building.

Quality, Legal and Security by Design



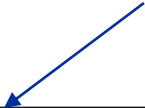
ADaM DataMindX AI





Example: ADEG.ANL01FL

1. Input IHP Specification



Dataset	Variable Name	Variable Label	Type	Algorithm
ADEG	ANL01FL	Analysis Flag	Char	If SCHVISFL='Y' , set ANL01FL=Y . If multiple scheduled measurements fall within the same USUBJID,PARAMCD,AVISITN group but were collected on different days, set ANL01FL=Y to the last measurement sorting by USUBJID PARAMCD AVISIT EGDTC EGSEQ.



Example: ADEG.ANL01FL

2. Revised Algorithm generated by AI

LLM Output - Revised Algorithm

<INPUT>=working_dataset

<VARIABLE>= ANL01FL

Algorithm:

1. Create working_dataset_subset by filtering records in working_dataset where SCHVISFL = 'Y'.
2. For each group defined by USUBJID, PARAMCD, and AVISIT in working_dataset_subset, identify measurements collected on different days (distinct EGDTC values).
3. Within each USUBJID, PARAMCD, AVISIT group in working_dataset_subset, sort records by EGDTC and EGSEQ in ascending order.
4. For each group, set ANL01FL = 'Y' for the last measurement (i.e., the record with the latest EGDTC and highest EGSEQ).
5. Merge the derived ANL01FL column from working_dataset_subset back into working_dataset using a left join, with working_dataset as the left dataframe.
6. In working_dataset, set ANL01FL = 'Y' for records identified in step 4; set ANL01FL = null for all other records.



Example: ADEG.ANL01FL

3.Final code generated by LLM

LLM2 Output – Generated R Code

```
adeg_eg_anl01fl <- function(working_dataset, output_variable = "ANL01FL") {  
  # Step 1: Filter records where SCHVISFL = 'Y'  
  working_dataset_subset <- working_dataset %>%  
    filter(SCHVISFL == "Y")  
  
  # Step 2: Identify measurements collected on different days (distinct EGDTC values)  
  # Step 3: Sort by EGDTC and EGSEQ within each group  
  # Step 4: Flag the last measurement in each group  
  working_dataset_subset <- working_dataset_subset %>%  
    group_by(USUBJID, PARAMCD, AVISIT) %>%  
    arrange(EGDTC, EGSEQ, .by_group = TRUE) %>%  
    mutate(  
      TEMP_FL = if_else(row_number() == n(), "Y", NA_character_)  
    ) %>%  
    ungroup()  
  
  # Step 5: Derive ANL01FL = 'Y' for last measurement, else NA  
  working_dataset_subset <- working_dataset_subset %>%  
    mutate(  
      !!sym(output_variable) := if_else(TEMP_FL == "Y", "Y", NA_character_)  
    ) %>%  
    select(USUBJID, PARAMCD, AVISIT, EGSEQ, !!sym(output_variable))  
}
```

```
# Step 6: Merge derived ANL01FL back into working_dataset  
working_dataset <- working_dataset %>%  
  left_join(  
    working_dataset_subset,  
    by = c("USUBJID", "PARAMCD", "AVISIT", "EGSEQ"),  
    suffix = c("", ".derived")  
  ) %>%  
  mutate(  
    !!sym(output_variable) := !!sym(paste0(output_variable, ".derived"))  
  ) %>%  
  select(-!!sym(paste0(output_variable, ".derived")))  
  
# Return the updated dataset  
return(working_dataset)  
}
```



Risk Mitigation & Trustworthy AI

-  **Cybersecurity** → Data encryption, Secure APIs.
-  **Bias Mitigation** → Curated pharma datasets + Human-in-the-loop checks.
-  **Drift Control** → Continuous monitoring, Model versioning.
-  **Enterprise Compliance** → Full audit logs, access control, red data never exposed.

Thank You