Building Successful ML Products

In [2019 Gartner predicted 80% of AI projects will remain as experiments run by individuals](https://blogs.gartner.com/andrew_white/2019/01/03/our-top-data-and-analytics-predicts-for-2019/). This is one of the key reasons why AI/ML Projects fail. Other reasons such as lack of data, talent, platforms, or leadership buy-in are more fundamental in nature, requiring a lot of time and effort to resolve. But once all the hard work is done, failing to operationalize models is undoing all hard work – management might get frustrated and stop sponsoring ML initiatives.

Machine learning projects typically start with a problem statement: A specific problem which can be solved using data and analytics. Data is collected and studied, and basic exploratory analysis is done. Data Scientists then run experiments and build models and train them. Upon validation, the model is frozen and outputs are generated. Most projects stop at this stage, and every time there is a need to generate output, the models are run manually

There are three major issues with this approach:

1. **Lack of Scalability:** Running models manually on local machines will not scale because of limitations in the size of machine. Also, as the scope of the model increases, size of data and steps involved in generating output increases, and hence running the steps manually becomes time consuming and infeasible
2. **Lack of Process Integration:** The end objective of any analytics project is to engrain analytics in business process/applications. Therefore, the whole end-to-end workflow, and integration has to be thought through upfront. The risk of model never being used is higher if workflow and integration become afterthought.
3. **Model Deterioration**: Any analytical model requires maintenance and constant monitoring of output. The quality of predictions deteriorate over a period of time. Model deterioration typically happens because of data drift (Changes in Data pattern over a period of time).

In order to overcome these issues, and increase chances of success, it is recommended to follow a product driven approach to analytics project. The end goal of a product driven approach is to build an Analytic product which is fully integrated with core applications, and something that goes through product lifecycle itself.

There are several approaches to build ML products and one such popular approach is the Drivetrain Approach <https://www.oreilly.com/radar/drivetrain-approach-data-products/>

Drive train approach is a top-down approach which is driven by business objectives. It seeks to add value to business processes by bringing data and insights into the process flow. Model building is just ' another brick in the wall'. Rather than being model centric, this approach strives to be Process and Data centric. Another important aspect of this approach is simulating the end result and the workflow. This is lacking in most of model centric approaches where the entire focus is on algorithms and accuracy metrics of the model. Drive-train approach provides equal weightage to

1. Business Process improvements
2. Application Integration
3. Ease of use and consumption
4. Models
5. MLOps

**The following is an extension of Drivetrain approach focusing on Analytical Product Design and Maintenance**

**Step 1:** *Clearly define an objective of the Product*. Eg: Recommendation engine that can drive sales by surprising and delighting customers.

**Key Tasks**

* Understanding Business Challenges and Key Business Processes impacted by the product
* AS-IS process
* Brainstorming ideas with Key Business Stakeholders (why do we need this product? What is the benefit? etc)
* Phrasing Objective Statements and sign off

***Tools (Platform)***

* MS Word/ Excel/Ppt

**Step 2:** ***Simulating the product:*** Identify workflows, inputs/outputs, interfaces, simulate end result, how will the outputs be consumed? Who are the consumers? How frequently will the product refresh present it to business obtain sign off

**Key Tasks**

* Identify Business Processes impacted
* Map process flow and identify Users(Actors), Inputs, Outputs, Processes
* Identify Analytics product touch points with the processes
* Develop Analytics product workflow (Model building 🡪 inferencing 🡪 Application integration)
* Define product SLAs - response time etc
* Define inferencing strategy (batch or real-time)
* Define UI/UX where applicable (Eg: process flow which requires users to upload data or trigger a run)

NOTE: At this point we have still not defined Model or improvements brought by it

***Tools (Platform)***

* MS Word/ Excel/Ppt
* Visio
* Jupyter notebook

**Step 3:** ***Identify levers of the product*** - What is the USP of the product? Eg: Ranking of recommendations, Sharp recommendations - at fit and size level with CTA and images of the product.

Key Tasks

* Define "Improvements" to be brought to the process very clearly. What constitutes significant improvement to existing process? What are the levers that can be used?
* What is the USP of this product?

***Tools (Platform)***

* MS Word/ Excel/Ppt

**Step 4:** *Gather relevant data needed to bring the improvements*

* Define what data are needed
* Identify Data Sources and Gather them (if data is not available see if they can be sourced afresh)
* Data Preparation
* EDA
* Insights
* Freeze all Data requirements, and Preparation

***Tools /Platform***

* MS Word/ Excel/Ppt
* SQL Workbench
* Jupyter notebook (Local machine / Azure ML)
* Azure ML

**Step 5** *Modelling - Select relevant models, experiment with multiple models*

* Select candidate models
* Simulate input and outputs of the model
* Prepare data for inputting into the model
* Start Experimentation
  + Feature extraction
  + Feature engineering
  + Model building
  + Model testing and validation
  + Model selection

***Tools /Platform***

* SQL Workbench
* Jupyter notebook (Local machine / Azure ML)
* Azure ML

**Step 6:** *MLOps*

* Deploy Pipelines to production
* Create Inferencing end points and share them with consuming applications
* Testing
* Schedule model run frequency
* Deploy model quality monitor and alerting mechanism
* Access control

***Tools /Platform***

* Azure ML

**Step 7:** *Complete Product Development and Rollout*

* Build UI of the product (Eg: interface for uploading data, or triggering a model run)
* Build Dashboards
* Deploy Solutions and roll out product

***Tools /Platform***

* Azure ML
* App development platform (for UI/UX , if required)
* Jupyter notebook