

GenericML

Look mah models without Containers, databricks, MLOps etc!

Challenges in AI/ML Implementation

- The vision of AI/ML has faced hurdles due to technical and communication barriers within organizations.
- Models are developed, deployed, tested and maintained by different teams and business users do not get to experiment/test models themselves.
- The usual problems of communication in software development apply. Agile development has not been standardised, many teams still struggle with agile development and waterfall development is sub-optimal.
- We need the ChatGPT style of model development for data problems.

Impedance in Model Deployment

- Data scientists often use tools like Python, R, and Julia, causing a disconnect with the development community who predominantly use languages like C#, Java, and Scala.
- This has caused Low Model Utilization Rates
- Studies show that 60-80% of models developed by data scientists never get into production mainly due to this technical impedance and also due to human factors such as the inability of domain experts to experiment with models.
- Technical issues hinder model deployment, including the lack of integration between data science tools and development environments. For example Domain Driven Design is not commonly used for application development let alone for data/reporting projects.

Importance of Iteration

- The ability to iterate on models is crucial for successful deployment.
- ML models usually overfit to their data, hence it is important for the business users to be able to select the parameters of models they think are going to be the most important and they need to experiment with the convergence level required for the models as some models with lower convergence may be good enough to give consistent results.
- Expensive MLOps using heavy compute is required due to the slow speed and high memory requirements of using Python. Hence the time available for experimentation is much less than it could be.

Automated Model Generation

- GenericML can be used to build tools for business users to select datasets, time frames, and parameters for automated model generation, eliminating the need for manual intervention.

Multiple Interfaces

- Multiple API interfaces can be provided for the generation and deployment of models. E.g. Rest, gRPC, WebSockets, NamedPipes, Serverless interfaces. Hence the architecture supports diverse technology stacks and deployment environments, enabling seamless integration with various cloud providers.

Data Architecture Consulting

- We can provide business assistance with the modern data warehouse concepts to enable real-time data flows for ML applications to avoid expensive ETL, ELT operations.
- Implementing Domain Driven Design simplifies event handling and aligns data architecture with business requirements.
- Successful implementation of AI/ML solutions requires educating and involving business stakeholders, addressing their non-functional requirements, and offering rapid incremental development.

Competitors

- There are really no competitors in the space of being able to provide tools for business users, i.e. non-programmers to train and test models themselves with an end to end solution in a fully dynamic fashion. There are tools that can be used in Jupiter notebooks e.g. PyCaret but these are only for the development of one off models. Models still need to be put into production and tested by non-technical users. Also due to the lack of abstraction in Python it is just not possible to provide a solution where the business can really play with selecting which parameters to use for a model and which variable will be the label (output variable) for a model for model predictions. This is possible with the framework as it C# uses generics and reflection.

Market Opportunity

- The market opportunity is enormous, just as ChatGPT has been so successful for text data it will be easy to provide custom tools for any business application that requires very little computing power.
- Business experts are just not aware of what is possible by switching data science to use languages from Python to more powerful type safe languages such as C#.
- Eventually full stack developers will also become AI/ML developers so the framework is a move in the right direction.
- Applications That Could Use GenericML
- GenericML is designed to bridge the gap between model development and deployment by abstracting the complexity of

traditional ML tools into a .NET-friendly framework. It supports full automation, domain-driven design, and vectorized data processing—making it ideal for a wide range of applications:

-
- Finance & Trading – Real-time trading bots, risk scoring, anomaly detection in transactions, dynamic pricing models.
- Healthcare – Disease risk prediction, anomaly detection in patient monitoring data, resource optimization for hospitals.
- Manufacturing – Predictive maintenance, defect detection, quality control optimization, yield forecasting.
- Utilities & Energy – Demand forecasting, power grid optimization, fault detection in sensor streams.
- Public Safety & Disaster Response – Fire risk prediction, flood modeling, early warning systems using satellite and sensor

data.

- Enterprise IT & Operations – Log analysis, fraud detection, resource allocation and optimization.
- E-Government & Public Sector – Social service fraud prediction, policy impact modeling, real-time infrastructure monitoring.
- Environmental Monitoring – Deforestation detection, wildlife movement clustering, pollution anomaly spotting, climate change monitoring, e.g. sea ice, bushfire detection/monitoring etc
- Education – Student dropout prediction, adaptive learning path models, academic risk analysis.
- Agriculture – Crop yield prediction, pest outbreak detection, soil health classification using sensor fusion.

- GenericML enables domain experts and developers to collaborate more effectively, bringing AI into mainstream .NET applications with minimal friction.

SECTION: ML Issues & GenericML Solutions

Solving ML Delivery Bottlenecks

Issues in traditional ML — and how GenericML fixes them

The Enterprise ML Reality Check

- Months-long model cycles, large infra bills, and brittle pipelines stall ROI.
- Data prep and leakage risks dominate effort, not model innovation.
- Business experts are out of the loop, so adoption and trust suffer.
- AutoML is too slow/expensive in common stacks; iteration speed is poor.

Issue #1 — Data Quality, Bias & Leakage

- Inconsistent sources and schemas lead to hidden selection bias.
- Temporal leakage and target bleed inflate offline metrics, fail in prod.
- Weak feature lineage makes audits, safety, and compliance hard.

GenericML Fix #1 — Data-Robust Modeling

- Schema-flexible vectors support single values and multi-dimensional arrays.
- Built-in temporal splits, leak checks, and repeatable feature pipelines.
- Fast re-runs let you A/B variants per domain slice to expose bias quickly.

Issue #2 — Overfitting & Brittle Models

- Hundreds of features + small, shifting regimes → unstable models.
- Grid/hand tuning is slow; models drift before deployment finishes.
- No simple way to inject domain rules or sanity constraints.

GenericML Fix #2 — Stable Generalization

- Automated feature selection, regularization, and robust validation schemes.
- Ensemble-ready: combine statistical signals with rules/constraints easily.
- Fast retraining on laptops enables frequent refresh against regime shifts.

Issue #3 — Cost & Iteration Speed

- GPU-centric pipelines and MLOps stacks slow teams and raise costs.
- AutoML in mainstream tools is too slow to be used continuously.
- Prototype-to-production gap causes duplicated work and re-writes.

GenericML Fix #3 — Fast, Low-Cost Iteration

- Train on GB-scale data locally without GPUs; cut infra by orders of magnitude.
- Auto feature engineering + model selection optimized for speed.
- Human-in-the-loop UX: domain experts can iterate safely and rapidly.

Issue #4 — Low Adoption & Explainability

- Stakeholders don't see how models behave; trust and uptake remain low.
- Hard to connect predictions to actions in existing applications.

GenericML Fix #4 — Human-Centered ML

- Transparent reports on features and contributions for each prediction.
- Plug-in summaries and guardrails expose trade-offs to business users.
- Embeddable components integrate into existing apps for decisions-at-point-of-use.

What This Means in Practice

- Weeks → days for first useful models; hours for new variants.
- Lower infra costs enable many more experiments and guardrail checks.
- Higher adoption: experts co-own model behavior and trust outputs.

How It Fits Your Stack

- Works alongside your data lake/warehouse; no rip-and-replace.
- Outputs portable models/artefacts; easy to schedule in existing orchestrators.
- Optional knowledge-graph integration for lineage, governance, and retrieval.

Next Steps

Pick two high-friction ML use-cases • Stand up a GenericML workspace • Timebox to 2–3 weeks