AI Powered Feature Engineering for Machine Learning Models

Ravi B. Sojitra

2024-06-04

ML models scale decision making everywhere.

Example of decision	Examples of predictions used for decision
Target: Discounts, promotions.	Lifetime value, gender, interests, location
Spend: Auto-bidding, pacing in ad auctions.	Likelihoods of conversion, click
Rank: Ads, brands, content, links.	Likelihoods of like, share; Dwell time/Time spent
Price: Products, services (Abnb, Amzn, Uber).	Likelihoods of Add-to-Cart, checkout, transaction
Match: Drivers and riders.	Likelihoods of requesting and accepting rides
Audit: Bots, policy violations, fraud.	Likelihoods of spam, topic; sentiment

Each automated decision may use 100s of model predictions!

So, modeling is automated to improve reliability.

All you have to do is select an outcome, an error metric, & a set of features and models to try.

1. Performance: Algorithms handle feature & model selection, (hyper)parameter tuning.
2. Deployment: You land the configuration and the system takes over logging, scheduling.
3. Monitoring: A tool will simply e-mail the on-call if something breaks.

Recent work[1] even aims to save you the hassle of looking for datasets and models to try.

1. Nicolo Fusi, Rishit Sheth, and Huseyn Melih Elibol (2018). Probabilistic matrix factorization for automated machine learning.

Challenge: Feature engineering is manual, slow.

Models often have low accuracy on simple classification tasks due to stale or few features.

Below, are some reasons for this.

- **Changes in privacy expectations**: E.g. retention and data use limits, Apple's policies.
- **Incentives**: It is difficult to evaluate the business impact of improving individual models.
- **Cost**: E.g. naive computation of graph/network statistics is slow and expensive.
- **Latency**: E.g. informative feature may take too long or may be expensive to compute.
- **Nonstationarity**: Underlying relationships between features and outcomes change.
- **Institutional knowledge**: It is difficult to find the right tables or information you need.

Proposal: Use AI to scale feature engineering.

Category	Example Tasks
Ideation	Identify features one would expect to be predictive of outcomes.
Institutional Knowledge	Determine row (“grain”) and column definitions of tables; Assess data log quality.
Compliance	Define purpose, data use, assess privacy commitments.
Preprocessing	Prototype features constructed using (un)structured data.
Engineering	Implement logging if necessary; Assess scalability of prototype; Improve efficiency.
Onboarding	Tooling, A/B testing, monitoring pipelines, launching changes.
R&D	Prototype embedding, graph sampling, optimization algorithms in scholarly work.

Evaluation: How I will know whether AI works.

Category	Example Tasks	Evaluation Metrics	Comparison
Ideation	Identify features one would expect to be predictive of outcomes.	Numbers of features defined, evaluated, launched as pipelines, model improvement	Features used for model fitting before vs after AI suggestion (per fit)
Institutional Knowledge	Determine row (“grain”) and column definitions of tables; Assess data log quality.	Hours/Days to find relevant upstream tables, program first prototype	Before versus after table and column auto-documentation
Compliance	Define purpose, data use, assess privacy commitments.	Weeks until legal & policy approval; Time verifying whether use case is already approved	Before vs after access to on-demand AI legal/policy office hours
Preprocessing	Prototype features constructed using (un)structured data.	Time to execute first feature prototype and assess impact on model performance	New employees onboarded to AI model first versus to SQL first
Engineering	Implement logging if necessary; Assess scalability of prototype; Improve efficiency.	Time to implement changes from code review; latency and costs (compute, storage) of launch	Users given vs not given beta access to AI code profilers, copilots
Onboarding	Tooling, A/B testing, monitoring pipelines, launching changes.	Time to complete task for which the tool is necessary; Time to resolve pipeline issues	Before versus after AI help (e.g. finding buttons) is introduced to tools
R&D	Prototype embedding, graph sampling, optimization algorithms in scholarly work.	Time to implement algorithm, test cases; Number of test cases implemented;	Before vs after access to AI fine tuned to code algorithms in papers

Thank you!