

PyMC-Marketing: Bayesian Marketing Mix Models and Customer Analytics in Python

William Dean ¹, Juan Orduz ¹, Colt Allen ¹, Carlos Eduardo Trujillo Agostini ¹, Ricardo Vieira ¹, Benjamin T. Vincent ², Thomas V. Wiecki ¹, Nathaniel Forde ¹, Luciano Paz ³, Pablo de Roque ¹, Imri Sofer ¹, Larry Dong ⁵, and Erik J. Ringen ⁴

¹ PyMC Labs, Estonia ² InferenceWorks Ltd, Scotland ³ PLC Tech d.o.o., Slovenia ⁴ Independent Researcher, UK ⁵ Dalla Lana School of Public Health, University of Toronto, Toronto, Canada

DOI: [10.21105/joss.10805](https://doi.org/10.21105/joss.10805)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Neea Rusch](#)  

Reviewers:

- [@allanvc](#)
- [@DanWaxman](#)
- [@markean](#)

Submitted: 14 May 2026

Published: 27 August 2026

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

PyMC-Marketing is a comprehensive Python library implementing Bayesian marketing analytics, built on PyMC ([Abril-Pla et al., 2023](#)). Commercial marketing analytics tools typically provide limited transparency into their models, while open-source alternatives like Meta's Robyn and Google's Meridian focus primarily on media mix modeling ([Google Inc., 2023](#); [Meta Platforms Inc., 2022](#)). PyMC-Marketing provides a unified framework spanning multiple marketing domains, including: Marketing Mix Models (MMMs), Customer Lifetime Value (CLV) analysis, Bass Diffusion Models, and Customer Choice Models. All outputs include full posterior distributions rather than point estimates, enabling explicit risk assessment in business decisions. Key capabilities include time-varying media coefficients via Hilbert space Gaussian process approximations ([Solín & Särkkä, 2020](#)), experimental calibration integrating lift tests into model likelihood, hierarchical Buy-Till-You-Die (BTYD) models ([Fader & Hardie, 2009](#)) for CLV, and budget optimization with business constraints. The library is available on PyPI and conda-forge, with comprehensive documentation and example notebooks at <https://www.pymc-marketing.io>.

Statement of Need

Marketing organizations struggle to attribute sales outcomes to specific marketing activities across multiple touchpoints and delayed conversion effects. Existing solutions suffer from: (1) black-box proprietary models with limited customization; (2) oversimplified approaches failing to capture marketing dynamics; and (3) lack of uncertainty quantification for high-stakes decisions. The target audience includes marketing data scientists and analysts in industry, academic researchers in causal inference and marketing science, and practitioners building production marketing measurement systems.

PyMC-Marketing addresses these gaps by bridging marketing science research and practical applications. It operationalizes Bayesian methods—hierarchical modeling, experimental calibration, and uncertainty quantification—within a user-friendly, scikit-learn-style API. Key innovations include time-varying coefficients using modern Gaussian process approximations optimized for marketing contexts, and a novel experimental calibration framework that integrates lift test results directly into model likelihood.

State of the Field

Existing marketing mix modeling tools include Meta's Robyn ([Meta Platforms Inc., 2022](#)) and Google's Meridian ([Google Inc., 2023](#)), which focus primarily on MMM with limited Bayesian inference capabilities. While Robyn provides bootstrap-based uncertainty intervals, it lacks full posterior distributions. Meridian offers Bayesian inference but is limited to MMM without extending to customer lifetime value, choice analysis, or product diffusion modeling. Benchmarks demonstrate that PyMC-Marketing achieves more efficient sampling and more accurate channel contribution recovery than Meridian, with explicit Fourier-based seasonality providing clearer separation of trend, seasonality, and media effects ([Säilynoja & Fiaschi, 2025](#)).

PyMC-Marketing fills this gap by providing a unified Bayesian framework across multiple marketing domains (MMM, CLV, Bass diffusion, choice modeling) with full uncertainty quantification. Contributing to existing tools was not appropriate for several reasons: Robyn is implemented in R with bootstrap-based inference, making Python-native full Bayesian methods architecturally incompatible; neither tool extends beyond MMM to CLV, choice modeling, or product diffusion. We therefore created a standalone Python library integrating Bayesian methods (hierarchical modeling, experimental calibration, time-varying parameters via modern GP approximations) within a scikit-learn-style API. This design enables both methodological research and production applications while maintaining computational efficiency.

Software Design

PyMC-Marketing follows a modular component architecture built on the PyMC probabilistic programming framework ([Abril-Pla et al., 2023](#)). The design prioritizes flexibility and extensibility through pluggable transformation components (adstock, saturation functions) and a builder pattern for model construction.

Key architectural decisions include: (1) separation of data transformation from model specification enabling custom function implementation; (2) a familiar fit/predict interface pattern lowering the barrier to adoption for practitioners; (3) PyMC backend providing automatic differentiation and multiple MCMC samplers; (4) standardized serialization for production deployment via MLflow ([Zaharia et al., 2018](#)). This architecture enables both methodological research and production applications while maintaining computational efficiency through GPU acceleration and modern sampling algorithms including NumPyro ([Bingham et al., 2019](#)) and Nutpie ([Seyboldt & PyMC Developers, 2023](#)). Quality is maintained through a comprehensive automated test suite integrated with GitHub Actions CI, covering core functionality across multiple Python versions.

Research Impact Statement

PyMC-Marketing provides uncertainty quantification through full posterior distributions, experimental calibration that grounds observational marketing models in causal estimates from lift tests, and flexible budget optimization with business constraints. The scikit-learn-style API ensures seamless integration into existing data science workflows. The library has been deployed in production by companies including HelloFresh ([PyMC Labs, 2023b](#)) and Bolt ([PyMC Labs, 2023a](#)). It is also featured in peer-reviewed surveys of open-source MMM tools ([Runge & Pauwels, 2026](#)).

Novel methodological contributions include: (1) time-varying coefficients using modern Gaussian process approximations specifically optimized for marketing applications; (2) experimental calibration framework integrating lift test results directly into model likelihood—a novel approach in the marketing science literature; (3) comprehensive marginal effects analysis for marketing sensitivity studies ([Arel-Bundock et al., 2024](#)). Comprehensive tutorials, example

notebooks, and video resources are available in the online documentation at <https://www.pymc-marketing.io/en/stable/>, with community support from over 70 contributors and translations in Spanish.

AI Usage Disclosure

Generative AI tools were used both in developing the software and in preparing this paper.

Software. PyMC-Marketing predates modern AI coding assistants, and its core model implementations were written by human contributors. More recently, contributors have used AI coding assistants, including Claude Code, Cursor, and GitHub Copilot, for implementation-level work such as drafting code and tests, writing docstrings and documentation, refactoring, and repository maintenance. Architectural design, API decisions, and statistical and methodological choices are made by the human maintainers. Every contribution, AI-assisted or not, enters the codebase through a pull request that requires approval by at least one human maintainer and must pass the automated test suite in continuous integration.

Paper. OpenCode (v1.0.220) with Claude Opus 4.5 and OpenCode (v1.14.48) with Claude Sonnet 4.6 via GitHub Copilot assisted with gathering information from existing documentation and codebase, drafting text, and incorporating peer reviewer feedback. All paper content was reviewed, edited, and validated by the human authors.

Acknowledgments

We acknowledge the PyMC development team for the foundational probabilistic programming framework and the broader PyData ecosystem contributors. Special recognition goes to the marketing science research community for developing the theoretical frameworks that PyMC-Marketing operationalizes (Bass, 1969; Fader et al., 2005; Jin et al., 2017; Schmittlein et al., 1987; Train, 2009).

PyMC-Marketing is a community-driven project built primarily through volunteer contributions from over 70 developers. Some contributions by PyMC Labs affiliates have received partial funding from both consulting engagements on marketing analytics and its internal budget. Larry Dong acknowledges support from the Dalla Lana School of Public Health, University of Toronto.

References

- Abril-Pla, O., Andreani, V., Carroll, C., Dong, L., Fonnesbeck, C. J., Kochurov, M., Kumar, R., Lao, J., Luhmann, C. C., Martin, O. A., Osthege, M., Vieira, R., Wiecki, T., & Zinkov, R. (2023). PyMC: A modern and comprehensive probabilistic programming framework in Python. *PeerJ Computer Science*, 9(e1516). <https://doi.org/10.7717/peerj-cs.1516>
- Arel-Bundock, V., Greifer, N., & Heiss, A. (2024). How to interpret statistical models using marginaeffects for R and Python. *Journal of Statistical Software*, 111(9), 1–32. <https://doi.org/10.18637/jss.v111.i09>
- Bass, F. M. (1969). A new product growth model for consumer durables. *Management Science*, 15(5), 215–227. <https://doi.org/10.1287/mnsc.15.5.215>
- Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P. A., Horsfall, P., & Goodman, N. D. (2019). Pyro: Deep universal probabilistic programming. *Journal of Machine Learning Research*, 20(28), 1–6. <https://jmlr.org/papers/v20/18-403.html>

- Fader, P. S., & Hardie, B. G. (2009). Probability models for customer-base analysis. *Journal of Interactive Marketing*, 23(1), 61–69. <https://doi.org/10.1016/j.intmar.2008.11.003>
- Fader, P. S., Hardie, B. G., & Lee, K. L. (2005). “Counting your customers” the easy way: An alternative to the Pareto/NBD model. *Marketing Science*, 24(2), 275–284. <https://doi.org/10.1287/mksc.1040.0098>
- Google Inc. (2023). *Meridian: Bayesian Media Mix Modeling at Scale* (Version 0.1.0). <https://github.com/google/meridian>
- Jin, Y., Wang, Y., Sun, Y., Chan, D., & Koehler, J. (2017). *Bayesian methods for media mix modeling with carryover and shape effects*. Google Inc. <https://research.google/pubs/pub46001/>
- Meta Platforms Inc. (2022). *Robyn: Marketing Mix Modeling* (Version 3.10.0). <https://github.com/facebookexperimental/Robyn>
- PyMC Labs. (2023a). *Better budgeting with Bayesian models: Bolt’s story with PyMC-Marketing*. <https://bolt.eu/en/blog/budgeting-with-bayesian-models-pymc-marketing/>
- PyMC Labs. (2023b). *Improving the Speed and Accuracy of Bayesian Media Mix Models for Marketing Optimization*. <https://www.pymc-labs.com/blog-posts/reducing-customer-acquisition-costs-how-we-helped-optimizing-hellofreshs-marketing-budget/>
- Runge, J., & Pauwels, K. (2026). Open-source media and marketing mix modeling: Practice-oriented overview, challenges, opportunities. *Customer Needs and Solutions*, 13(1). <https://doi.org/10.1007/s40547-026-00161-4>
- Säilynoja, T., & Fiaschi, L. (2025). *PyMC-Marketing and Meridian Revisited: Approaches to Baseline Modeling for MMMs*. <https://www.pymc-labs.com/blog-posts/pymc-marketing-vs-meridian-baseline-modeling-mmm>
- Schmittlein, D. C., Morrison, D. G., & Colombo, R. (1987). Counting your customers: Who are they and what will they do next? *Management Science*, 33(1), 1–24. <https://doi.org/10.1287/mnsc.33.1.1>
- Seyboldt, A., & PyMC Developers. (2023). *Nutpie* (Version 0.5.0). <https://github.com/pymc-devs/nutpie>
- Solin, A., & Särkkä, S. (2020). Hilbert space methods for reduced-rank Gaussian process regression. *Statistics and Computing*, 30(2), 419–446. <https://doi.org/10.1007/s11222-019-09886-w>
- Train, K. E. (2009). *Discrete choice methods with simulation*. Cambridge University Press. <https://doi.org/10.1017/CBO9780511805271>
- Zaharia, M. A., Chen, A., Davidson, A., Ghodsi, A., Hong, S. A., Konwinski, A., Murching, S., Nykodym, T., Ogilvie, P., Parkhe, M., Xie, F., & Zumar, C. (2018). Accelerating the Machine Learning Lifecycle with MLflow. *IEEE Data Eng. Bull.*, 41, 39–45. <http://sites.computer.org/debull/A18dec/p39.pdf>