International Journal of Emerging Trends in Computer Science and Information Technology
ISSN: 3050-9246 | https://doi.org/10.56472/ICCSAIML25-114
Eureka Vision Publication | ICCSAIML'25-Conference Proceeding

Original Article

No-Code vs Traditional Machine Learning for Lead Generation: A Comparative Case Study

Sai Yellaiah Simhadri
Senior Data Architect, Illinois, USA.

***Abstract** - This research paper compares no-code machine learning (ML) platforms, specifically AWS SageMaker Canvas, with traditional Python-based ML methods in the context of lead generation for a direct selling company. The study examines each approach's performance, cost-effectiveness, ease of use, effort and compatibility for various business scenarios using a real case study of a direct selling company with 40000 consultants and 10 million customers. It reveals that while traditional ML delivers improved performance (22% higher conversion improvement), it also demands specialized skills, significant development time and pipeline management. On the other hand, no-code solution offers faster implementation (12 vs. 18 weeks) and higher ROI (2566.67% vs 388.14%), while also enabling business users with minimal technical background to build and deploy predictive models efficiently. This research also helps companies to make informed decisions about their ML strategy and implementation for lead generation. Based on the findings, the study recommends using No-code ML platforms when speed, ease of use, and lower costs are prioritized; opting for traditional ML methods when business needs demand high customization, advanced analytics, and detailed model transparency and considering a hybrid approach to leverage the strengths of both solutions by prototyping quickly with no-code tools and deploying robust, scalable solutions using traditional ML techniques.[1][2]*

***Keywords** - AWS Sage Maker Canvas, Direct Selling, Lead Generation, Machine Learning, No-Code ML, Traditional ML*

1. Introduction

In today's highly competitive business world, finding new customers is very crucial for direct selling companies and in fact for any business. Companies in direct selling especially face a massive challenge: they have thousands of consultants and millions of potential customers. These companies rely on their consultants to build relationships and sell products directly to consumers. Consultants are basically their storefronts. Very quickly, they find it difficult to find new customers after reaching out to their inner circle, friends and close relatives. Many consultants don't have experience with social media marketing or finding time to prospect new leads which can impact their earnings and ultimately company's revenue as well.

To truly empower consultants, companies need to go beyond product training and provide necessary tools, platform for consultants to tap into new leads. For example, a simple smart lead scoring dashboard can provide insights and suggest contacts based on engagement levels, previous interactions and even customize product recommendations for each lead. With the rise of data and technology, companies can take support even further by leveraging machine learning to help consultants identify high potential leads more effectively. Using ML can make a big impact but it raises a key question: what's the right approach?" This paper looks at both approaches for a direct selling company with 40,000 consultants and 10 million customers.

We compare AWS SageMaker Canvas (a no-code tool)[2] with traditional Python-based ML[6] methods to see which works better for lead generation. We examine how each approach performs, what resources they require, how much they cost, and how easy they are for consultants to use. By comparing these approaches through real-world example, this paper helps direct selling companies choose the right ML strategy for their specific needs by comparing factors like performance, cost, implementation time. We look at when no-code tools might be the better choice, when traditional programming makes more sense, and how companies might combine both approaches for the best results.

2. Literature Review and Background

2.1 Early Traditional Methods vs Data-Driven Approaches

Early traditional lead generation in direct selling focuses on relationship-building through personal recommendations, community events, and product demonstrations. These methods create trust between consultants and customers, which is important in direct selling.[4] However, they're limited in how many people they can reach and how efficiently they work at a large scale.[7]

Data-driven approaches use customer information, behavior patterns, and market trends to identify and prioritize leads based on how likely they are to buy. These methods help consultants focus on the most promising prospects. Early data approaches used simple analysis like how recently and frequently customers bought products.[3]

The main difference between early traditional and data-driven methods is in decision-making. Early traditional methods rely on consultant intuition and experience, while data-driven approaches use objective measurements and systematic analysis. The best strategies usually combine both approaches.[7]

2.2 Machine Learning in Lead Generation – a Data-Driven approach

Machine learning has dramatically improved lead generation compared to earlier data methods. Unlike systems with fixed rules, ML algorithms can find complex patterns in customer data, adapt to changing behaviors, and continuously improve their accuracy over time. Machine learning has transformed lead generation in several important ways with Predictive Lead Scoring, Customer Grouping, Behavior Analysis, Predicting Customer Loss, Lifetime Value Prediction[3][5].

For direct selling companies, these capabilities help match the right consultant with the right prospect at the right time. By automatically analyzing complex customer data, machine learning enables more efficient distribution and personalized engagement at scale.[4]

2.3 The Rise of No-Code ML Platforms

Initially, using machine learning required specialized experts with advanced knowledge of statistics, programming, and business understanding. These data scientists were essential but hard to find, which limited who could use ML technology. No-code machine learning platforms emerged to solve this problem. These platforms make ML accessible through visual interfaces and automated processes. They hide the technical complexities, allowing business users with industry knowledge but limited technical skills to build and use ML solutions.[10]

AWS SageMaker Canvas is an example of this approach, providing a visual interface for building machine learning models without writing code. Some other offerings come from Google (AutoML), Microsoft (Azure Machine Learning), and other specialized platforms.[2]

2.4 Challenges in Lead Generation for Large Direct Selling Companies

Direct selling companies with large consultant networks and customer bases face several unique challenges:[4]

1. Scale and Complexity: Managing lead generation for 40,000 consultants serving 10 million customers creates enormous amounts of data and complex distribution needs.
2. Consultant Differences: Individual consultants have varying levels of experience, technical skills, and business knowledge, requiring lead generation systems that can work for everyone.
3. Scattered Data: Customer information often exists in multiple systems, including CRM platforms, e-commerce sites, social media, and consultant records, making it hard to get a complete picture.[11]
4. Personalization at Scale: Maintaining the personal touch that makes direct selling special while operating at enterprise scale requires sophisticated approaches to personalization.[11]

These challenges create complex decision making for direct selling companies considering machine learning for lead generation. The choice between no-code and traditional ML approaches should address these industry-specific factors while balancing technical capabilities, resource requirements, and implementation timelines.

2.5 No-Code Machine Learning Approach

2.5.1 Overview of AWS Sage Maker Canvas

Amazon SageMaker Canvas is a tool that makes machine learning accessible to people without coding skills. It's part of Amazon Web Services (AWS) and provides a visual interface that lets business analysts and subject matter experts build, train, and use machine learning models without writing any code.[2]

2.5.2 Key Features and Capabilities

SageMaker Canvas offers several features designed to simplify the machine learning process.

- Visual Interface and Workflow: The platform has an easy-to-use drag-and-drop interface that guides users through the machine learning process.[2]
- Built-in Models and Algorithms: SageMaker Canvas includes pre-built models for common business needs, including classification, regression, forecasting, and image recognition tasks. These models incorporate industry best practices and are optimized for performance, eliminating the need for users to understand algorithm selection or parameter tuning.[2]

- Data Preparation and Transformation Tools: The platform includes tools for cleaning data, transforming it, and creating useful features. Users can handle missing values, convert data types, create derived features, and perform other essential data preparation tasks through a point-and-click interface.[2]
- Integration with AWS Services: As part of the AWS ecosystem, SageMaker Canvas works seamlessly with other AWS services, including Amazon S3 for data storage, AWS Glue for data cataloging, and Amazon QuickSight for visualization.[2]

2.5.3 Advantages of No-Code ML for Lead Generation

The no-code approach offered by SageMaker Canvas provides several distinct advantages for lead generation in direct selling companies:

- Accessibility for Non-Technical Users: Perhaps the biggest advantage is making ML capabilities available to more people due to its no code nature.[10]
- Speed of Implementation: No-code platforms dramatically reduce the time needed to develop and deploy ML solutions. What might take weeks or months with traditional ML approaches can often be accomplished in days with SageMaker Canvas.
- Reduced Technical Overhead: No-code platforms minimize the technical infrastructure and maintenance requirements associated with ML implementations.[2]

2.5.4 Limitations of No-Code ML

Despite its advantages, the no-code approach has several limitations that must be considered:

- Customization Constraints: While SageMaker Canvas offers flexibility within its framework, it cannot accommodate highly specialized algorithms or novel approaches that might be required for complex lead generation scenarios. The platform is limited to the models, transformations, and workflows that have been pre-built into the system, which may not address unique business requirements due to its Back-box nature.[2]
- Potential Scalability Issues: While SageMaker Canvas can handle substantial data volumes, organizations with extremely large datasets or complex real-time processing requirements may encounter performance limitations.

2.6 Traditional Python-Based Machine Learning Approach

2.6.1 Overview of Python ML for Lead Generation

Python has become the most popular programming language for machine learning, offering many tools that help create sophisticated lead generation solutions. This coding framework gives data scientists and ML engineers powerful capabilities for handling data, building models, and putting them into use, allowing for highly customized approaches to find potential leads with better conversion.[12]

2.6.2 Key Libraries and Frameworks

The Python ML framework includes several important components that are particularly useful for lead generation:

- Scikit-learn: This versatile library provides many machine learning algorithms, including classification, regression, and clustering methods that work well for lead scoring and customer segmentation. [9]
- Pandas: As the main data handling library in Python, Pandas makes it possible to perform complex data transformations needed for feature engineering in lead generation models.[12]
- NumPy: This fundamental numerical computing library supports most Python ML frameworks, providing efficient array operations and mathematical functions.[12]
- TensorFlow and PyTorch: These deep learning frameworks allow the development of neural network models that can capture complex relationships in customer data.[10]

2.6.3 Advantages of Traditional ML for Lead Generation

The traditional Python-based approach offers several distinct advantages for lead generation applications:

- Full Customization Capabilities: Perhaps the most significant advantage is the unlimited flexibility to customize models to specific business requirements. Data scientists can implement custom algorithms, loss functions, and evaluation metrics that precisely align with the organization's definition of lead quality and conversion objectives.[6]
- Transparency and Explainability: Traditional ML approaches allow for complete visibility into the model development process and decision-making logic.[12]
- Advanced Feature Engineering: For direct selling companies where the customer journey involves multiple touchpoints across online and offline channels, this capability to engineer complex features is particularly valuable.[6]

2.6.4 Limitations of Traditional ML

Despite its advantages, the traditional ML approach presents several challenges:

- Technical Expertise Requirements: The most significant limitation is the need for specialized data science talent to develop effective Python-based ML solutions.[10]

- Development Time and Resources: Traditional ML projects typically require months of development time.[6]
- Maintenance Complexity: Custom ML solutions require ongoing maintenance to handle data drift, system updates, and evolving business requirements.
- Deployment Challenges: Implementing ML models in production environments involves complex DevOps considerations, including containerization, scaling, monitoring, and failover mechanisms.[12]

3. Results and Discussion

3.1 Case Study: Lead Generation for a Direct Selling Company

- Company Profile: 40,000 Consultants with 10 million Customer Base: The study focuses on a direct selling company in the kitchen tools industry, which has 40,000 consultants serving 10 million customers with over 500 different products. The business operates within a complex multi-level marketing structure and requires scalable lead generation solutions to efficiently prioritize high-value prospects.[4]
- Methodology: The comparative analysis examines two primary ML approaches:
  - No-Code ML using AWS SageMaker Canvas.
  - Traditional Python based ML development.

Evaluation criteria include performance metrics, ease of implementation, scalability, customization capabilities, and cost analysis.

- Implementation Details
  - No-code ML Approach: AWS SageMaker Canvas, utilizing pre-built ML models with automatic optimization. The model used is a two-category prediction using the XGBoost algorithm, deployed via AWS SageMaker Endpoint. This approach enables rapid deployment and accessibility for non-technical business users.[2]
  - Traditional ML Approach: The traditional method involves custom Python development, leveraging specialized data engineering and ML libraries. It uses a similar two-category XGBoost model without hyperparameter tuning, deployed through AWS SageMaker Endpoint. This approach offers greater customization, flexibility, and transparency, though it requires extensive technical expertise.[6]

3.2 Comparative Analysis

The evaluation was performed on datasets of 100,000 customer predictions:

3.2.1 Implementation time Comparison

Table 1. Resource Requirements Comparison

	No-Code	**Traditional**
Phase 1: Data Preparation	4 weeks	5 weeks
Phase 2: Python code development	N/A	4 weeks
Phase 3: Model development	2 weeks	2 weeks
Phase 4: Integration & Deployment	4 weeks	5 weeks
Phase 5: Rollout and Training	2 weeks	2 weeks
Total	12 Weeks	18 Weeks

The total implementation timeline for the traditional ML approach was approximately 18 weeks, 6 weeks more than no-code implementation.

3.2.2 Performance Metrics Comparison

Table 2. Model Quality metrics comparison

	No-Code		**Traditional**	
Accuracy	**88.11%**		**88%**	
	0	**1**	**0**	**1**
Precision	86.17%	90.17%	85%	90%
Recall	90.38%	85.88%	89%	86%
F1-Score	88.22%	87.97%	87%	88%

Both approaches achieved comparable accuracy levels with No-Code offering slighlty better accuracy and Recall metrics.

3.2.3 Resource Requirements

Table 3. Resource requirements comparison

	No-Code	**Traditional**
Resources	2.25 full time staff	4.5 full time staff

Traditional approach requires 2x more personnel for implementation.

3.2.4 Conversion rate comparison

Conversion rate is compared against the base conversion rate of **0.7%,** which is through non data-driven approach relying on consultant intuition and marketing efforts.

Table 4. Conversion rate comparison

	No-Code	Traditional
Conversion rate	300% improvement (0.7% > 2.8%)	322% improvement (0.7% > 2.96%)

Traditional approach delivered 22% higher conversion rate improvement.

3.2.5 Costs comparison

Table 5. Costs comparison

	No-Code	Traditional
Development costs	~$18000	~$118000
3-Year Total cost of ownership	~$54000	~$354000

Traditional approach costs **6.55x more** to develop and maintain.

3.2.6 Return on Investment

While both approaches delivered strong returns, the no-code solution provided a significantly higher ROI percentage due to its lower upfront investment in development resources, infrastructure, and specialized talent. Although the traditional approach generated a higher absolute incremental profit by delivering more precise predictions and improved conversion rates, it also incurred substantially greater costs in terms of implementation time, personnel, and ongoing maintenance.

Table 6. ROI comparison

	No-Code	Traditional
Estimated Annual Incremental Revenue	~$1.6M	~$1.92M
Estimated Annual Incremental Profit	~$480000	~$576000
ROI	2566.67%	388.14%

3.3 Discussion

3.3.1. Key findings from comparative Analysis

Our comparison of no-code and traditional machine learning approach for lead generation in the above case study reveals several important findings that can help similar organizations make better decisions. First, we found a clear trade-off between performance and investment. The traditional ML approach delivered better predictive performance, with about 22% higher improvement in lead conversion rates and better precision and recall metrics. However, this performance advantage came at a much higher cost, requiring 6.55x times the development investment and total cost over three years. This raises a basic question for organizations: is the better performance really worth the much higher investment? Second, the time-to-value difference between both approaches is substantial.

The no-code implementation delivered results 6 weeks earlier (12 vs. 18 weeks) compared to the traditional approach. This faster timeline could be a big advantage, even if it comes with some performance compromises. Third, resource requirements and technical expertise needed for each approach differ dramatically [10]. The traditional ML approach required nearly twice the full-time equivalent resources and demand specialized data science and ML engineering skills.

3.3.2. Scenarios where No-Code ML excels for Lead generation

Based on our analysis, several scenarios emerge where the No-Code ML approach is the optimal choice for lead generation.

- Organizations with limited budgets or data science resources, No-code ML is a practical and cost-effective option.
- The ability to deliver results faster enables organizations to realize business benefits more quickly and adapt faster.
- When the lead generation requirements align well with standard classification and regression models without requiring extensive customization or specialized algorithms, no-code platforms can deliver sufficient performance without the overhead of custom development [3].

3.3.3. Scenarios where Traditional ML is preferable

Conversely, several scenarios exist where traditional python ML approach is the optimal choice despite their higher cost and complexity [6].

- Performance critical applications where slight improvement in conversion rates has substantial financial impact which justifies higher investment.
- Organizations with complexities in data with unique customer journey patterns and multi-level compensation

structures may require customization capabilities of traditional ML [6].

- Companies that have already invested in data science teams and ML infrastructure may find traditional approach more aligned with their existing capabilities and able to leverage resources more effectively [10].

3.3.4. Hybrid Approach possibilities

Our analysis suggests that the choice between no-code and traditional ML need not be an either/or decision. Several hybrid approaches could combine elements of both methodologies to optimize the balance between performance, investment, and time-to-value [10] [8]:

- Organizations could begin with no-code approaches to deliver quick wins and establish proof of concept, then selectively transition high-value or complex components to traditional ML implementations as the business case justifies the additional investment [8].
- Different lead generation scenarios within the same organization might be best served by different approaches. For example, basic lead scoring for new customer acquisitions might use no-code solutions, while more complex consultant recruitment prediction might leverage traditional ML for its superior customization capabilities [8].

4. Conclusion

No-code ML platforms offer clear advantages such as faster implementation, lower technical barriers, and accessibility for non-technical users, making them ideal for rapid deployment. In contrast, traditional ML methods excel in areas like advanced feature engineering, deep customization, and model transparency, making them better suited for businesses that require high precision and control. The decision between the two approaches should be based on the organization's specific context, including its maturity, technical capability, and business goals.

While there is no one-size-fits-all solution, no-code ML can deliver around 90% of the value at just 20% of the cost, making it a highly efficient option for many scenarios. Traditional ML, though more resource-intensive, can offer superior performance when needed. A hybrid strategy often provides the best of both worlds rapid prototyping through no-code platforms, followed by refining and deploying production-ready solutions using traditional ML techniques. Choosing the right ML path ensures organizations achieve the best results aligned with their needs and capabilities.

5. Conflicts of Interest

The author declare that there is no conflict of interest concerning the publishing of this paper.

References

[1] Internal company case study data and implementation experiences.
[2] Amazon Web Services. (2024). No-code Machine Learning - Amazon SageMaker Canvas. Retrieved from https://aws.amazon.com/sagemaker-ai/canvas/
[3] Gupta, S. (2024, February 19). A Deep Dive into Lead Scoring Techniques Using Machine Learning. Medium. Retrieved from https://medium.com/@creatorvision03/a-deep-dive-into-lead-scoring-techniques-using-machine-learning-3ec85864645a
[4] Integrated MLM Software. (2025, March 28). AI in MLM: How Artificial Intelligence is Transforming Direct Sales. Retrieved from https://integratedmlmsoftware.com/artificial-intelligence-direct-selling-mlm/
[5] Tpoint Tech. (2024). Lead Generation using Machine Learning. Retrieved from https://www.tpointtech.com/lead-generation-using-machine-learning
[6] Szabo-Toke, A. (2023, August 30). How I built an AI lead scoring model with Python. LinkedIn. Retrieved from https://www.linkedin.com/pulse/how-i-built-ai-lead-scoring-model-python-adam-szabo-toke
[7] Dasha.AI. (2023, October 24). Generative AI vs. Traditional Methods in Lead Generation. Retrieved from https://dasha.ai/en-us/blog/generative-ai-vs-traditional-methods-in-lead-generation
[8] Cognism. (2025, March 18). AI for B2B Lead Generation: The Ultimate Guide. Retrieved from https://www.cognism.com/blog/ai-lead-generation.
[9] Scikit-learn: Machine Learning in Python – https://scikit-learn.org/stable/
[10] McKinsey & Company. (2024). The State of AI in 2024: Generative AI's Breakout Year. McKinsey Digital.
[11] Salesforce. (2024). State of the Connected Customer Report.
[12] R. Daruvuri, "An improved AI framework for automating data analysis," World Journal of Advanced Research and Reviews, vol. 13, no. 1, pp. 863–866, Jan. 2022, doi: 10.30574/wjarr.2022.13.1.0749.
[13] Brownlee, J. (2024). Your First Machine Learning Project in Python Step-By-Step. Machine Learning Mastery. Retrieved from https://www.machinelearningmastery.com/machine-learning-in-python-step-by-step/