Staff Machine Learning Engineer - Take Home Assignment

Marketing AI/ML Algorithms & Applications Team

Thank you for participating in the interview process with Salesforce! We are excited to see how you tackle machine learning engineering challenges and build production ML systems.

Instructions

This assignment presents two ML system engineering challenges. You may choose to implement either one or both parts, depending on your time and interests. Please complete your chosen scope within 5 days of receiving this assignment. While we suggest 4-6 hours of effort, you're welcome to spend more time if you wish. After submission, you may be invited to present your solution to a panel of Salesforce employees.

Background

The Marketing AI/ML Algorithms & Applications team is developing a wide range of predictive and inferential models that address key marketing effectiveness measurement and optimization use cases for our B2B business. We develop propensity models, net-lift models, and recommendation engines to optimize outcomes of marketing campaigns. We also create complex inferential models that estimate the value of each marketing program/campaign on revenue/ACV generation to measure the effectiveness of marketing spend.

We use a wide range of AI/ML technologies, including:

- Statistical models (e.g., logistic and linear regressions)
- Neural networks (e.g., ANN, RNN, LSTM)
- Tree-based algorithms (e.g., XGBoost, LightGBM, Random Forest)
- Causal models (e.g., CausalML)
- Unsupervised clustering
- Decision boundaries algorithms (e.g., SVM)

We build our algorithms at a contact, lead, opportunity, and account levels (remember, we are a B2B company).

We use AWS and Snowflake as our primary data and AI/ML infrastructure. The input data for our models come from a variety of sources:

- Contact/lead/opportunity/account attributes and interactions/touchpoints with Salesforce marketing and sales come from a Snowflake database and a data lake
- Contact/lead website browsing activity comes from the Google Analytics platform

- Contact/lead historical form fill data and email communications from AWS S3

Implementation Notes

- Your solution should be fully implemented and runnable
- Include captured output/logs demonstrating successful execution
- Be prepared to run a live demo during the panel interview
- Use your personal AWS account for implementation (you'll need to create required resources)
- If AWS cost is a concern, document your complete design and implement a meaningful subset
- Focus on quality over quantity - a well-implemented partial solution is better than a rushed complete solution

Option 1: Model Serving Infrastructure

Real-time Lead Scoring System

In this hypothetical scenario, we have a lead scoring model in production that needs significant infrastructure improvements. The current system:

- Uses XGBoost to score leads from 1-5
- Processes ~300 requests/second
- Must return scores within 1 second
- Runs on AWS SageMaker
- Uses 50 input features from various sources
- Writes results to our data lake

Build a production-ready serving infrastructure that demonstrates your understanding of ML systems in production. Key components to consider:

1. A scalable REST API service

2. Infrastructure setup (choose appropriate components):

    - Load balancing
    - Auto-scaling
    - Monitoring
    - Logging
    - Security

3. CI/CD pipeline elements such as:

    - Testing strategy

- Deployment automation
- Security checks

4. Monitoring capabilities like:

    - System metrics
    - ML-specific metrics
    - Alerting

5. Documentation of:

    - Architecture
    - Operations
    - Deployment
    - Monitoring

Option 2: Model Retraining Pipeline

Automated Model Updates

In this hypothetical scenario, our lead scoring model needs regular retraining as data patterns change. Build an automated retraining pipeline focusing on key MLOps capabilities:

1. Core retraining functionality:

    - Training orchestration
    - Model validation
    - Version control
    - Deployment strategy

2. Infrastructure components such as:

    - Training environment
    - Model registry
    - Validation setup

3. Quality assurance including:

    - Data validation
    - Performance metrics
    - Impact analysis

4. Operational monitoring:

- Pipeline status
- Resource usage
- Cost tracking

Deliverables

For your chosen part(s), provide:

1. Working code in a Git repository
2. Infrastructure configurations
3. Pipeline definitions
4. System documentation
5. Test coverage
6. Monitoring setup
7. Execution logs/output demonstrating successful runs
8. README with:
    - Setup instructions
    - Run instructions
    - Design decisions
    - Scope decisions
    - Future improvements

Notes

- Prioritize robust, production-quality implementation of core functionality
- Make deliberate scope decisions and document your reasoning
- Consider security, scalability, and maintainability
- Include error handling and logging
- Focus on ML system engineering best practices

If you have questions about requirements or scope, please reach out to your recruiter.