AI Agents in the Smart Notification System

Each agent in our system will serve a specialized purpose with specific AI capabilities:

1. User Behavior Analysis Agent

Purpose: Analyze individual user interaction patterns to build a comprehensive profile of preferences and behaviors.

AI Implementation:

- **Algorithm:** Primarily using sequence modeling (LSTM/Transformer models) to understand temporal patterns in user behavior
- **Features:** Session frequency, activity duration, click patterns, time-of-day preferences
- **Output:** User engagement profile with temporal patterns and preference scoring

POC Implementation:

- Simplified version using rule-based analysis with basic ML classification
- Track and analyze 3-4 key metrics (visit frequency, time spent, click-through rates)
- Deploy as containerized microservice with REST API interface

2. Content Preference Agent

Purpose: Determine which notification types (shipment, packing, delivery) are most relevant to each user.

AI Implementation:

- **Algorithm:** Collaborative filtering combined with content-based recommendation systems
- **Features:** Historical engagement with notification types, contextual business state, user segment characteristics
- **Output:** Ranked list of notification types with relevance scores for each user

POC Implementation:

- Basic collaborative filtering algorithm analyzing notification open/click patterns
- Track engagement by notification category
- Simple content-based features using notification metadata

3. Channel Optimization Agent

Purpose: Determine the optimal delivery channel (email, mobile, SMS, dashboard) for each notification.

AI Implementation:

- **Algorithm:** Multi-armed bandit reinforcement learning algorithm
- **Features:** Channel-specific response rates, time-to-response by channel, device usage patterns
- **Output:** Channel selection probability distribution with confidence scores

POC Implementation:

- Thompson sampling algorithm for email vs. dashboard optimization
- Focus on just two channels for POC (email and dashboard)
- Track response time and engagement rates by channel

4. Frequency Optimization Agent

Purpose: Determine optimal notification frequency to maximize engagement while minimizing notification fatigue.

AI Implementation:

- **Algorithm:** Gaussian Process model with Bayesian optimization
- **Features:** Historical engagement decay curves, notification density tolerance, time-sensitivity of content
- **Output:** Optimal notification interval ranges with confidence bounds

POC Implementation:

- Simple Bayesian model tracking engagement decay with notification frequency
- A/B testing between standard fixed frequency and adaptive frequency
- Monitor unsubscribe/mute rates as fatigue indicators

5. Orchestration Agent

Purpose: Coordinate between all specialized agents to make final notification decisions.

AI Implementation:

- **Algorithm:** Ensemble decision model with weighted agent outputs
- **Features:** Outputs from all specialized agents, business rules, priority frameworks
- **Output:** Final notification decisions with timing, channel, and content prioritization

POC Implementation:

- Rule-based orchestration with simple weighting of agent recommendations
- Configuration interface for business rule integration
- Decision logging for optimization analysis

Agent Implementation in Project Timeline (For Proper Implementation)

Let me update the POC sprint plan to specifically highlight the agent implementation:

Sprint 1: Project Setup & Basic Agent Framework (April 21 - May 4, 2025)

- Set up agent communication framework and API interfaces
- Define agent input/output schemas and interaction patterns
- Implement basic versions of User Behavior Analysis Agent and Content Preference Agent

Sprint 2: Core Agents Development (May 5 - May 18, 2025)

- Develop Channel Optimization Agent with A/B testing framework
- Implement Frequency Optimization Agent with basic learning capabilities
- Create simplified Orchestration Agent with rule-based decision making
- Build agent monitoring dashboard to visualize agent decisions

Sprint 3: Agent Testing & Refinement (May 19 - June 1, 2025)

- Test agent performance with synthetic and initial real user data
- Refine agent algorithms based on preliminary results
- Implement feedback loops for agent learning
- Add logging and explainability features to agent decisions

Key Technical Components for AI Agents

1. **Agent Framework**
    - Containerized microservices for each agent with standard API interfaces
    - Event-driven architecture for real-time agent cooperation
    - State management system for agent learning and adaptation
2. **ML Infrastructure**
    - Feature store for consistent agent input processing
    - Model registry for version control and deployment
    - Experiment tracking system for agent performance comparison
3. **Agent Feedback Loop**
    - Real-time metric collection for reinforcement learning
    - A/B testing framework for agent strategy comparison

    - Automated model retraining pipeline based on performance metrics

4. **Observability for Agents**
    - Decision logging for each agent with reasoning
    - Performance dashboards for agent effectiveness
    - Anomaly detection for agent behavior

The AI agents form the intelligent core of our notification system, moving beyond simple rules to truly adaptive behavior that improves over time. Each agent specializes in a different aspect of the notification problem, working together through the orchestration agent to deliver a cohesive, personalized notification experience.