Project Proposal

Prediction Market Analysis System

David Yoder

January 2026

Course: Independent Study: Semantic Information Systems
Instructor: Prof. Owen Mundy
Duration: 2 Weeks

1 Problem Statement

Prediction markets like Kalshi aggregate collective beliefs about future events into tradeable prices. These markets theoretically incorporate all publicly available information, but several inefficiencies exist:

- **Information asymmetry:** Price movements often lag publicly available signals from news, official announcements, and other sources. The speed at which markets incorporate new information varies significantly.
- **Manual monitoring is impractical:** Hundreds of active markets exist on the platform, each with different resolution criteria and catalysts. Tracking relevant developments manually does not scale.
- **Pattern opacity:** Volume spikes, order book changes, and price movements contain information, but without systematic collection and analysis, these patterns remain difficult to identify and exploit.

Existing tools focus on either fully autonomous trading (high risk, requires significant capital) or static dashboards (miss real-time opportunities). There is a gap for systems that surface actionable insights for human decision-making without requiring automated execution.

2 Goal

Build an exploratory system that monitors Kalshi prediction markets, correlates price movements with external signals, and surfaces potential inefficiencies. The primary focus is learning and analysis rather than profit extraction. Specifically, the system will:

1. Continuously collect price, volume, and order book data from the Kalshi API
2. Aggregate external signals from free sources (RSS feeds, Reddit, market activity patterns)
3. Detect and alert on interesting patterns: price movements without corresponding signals, signals without price response, unusual volume activity

4. Document findings to answer research questions about market efficiency

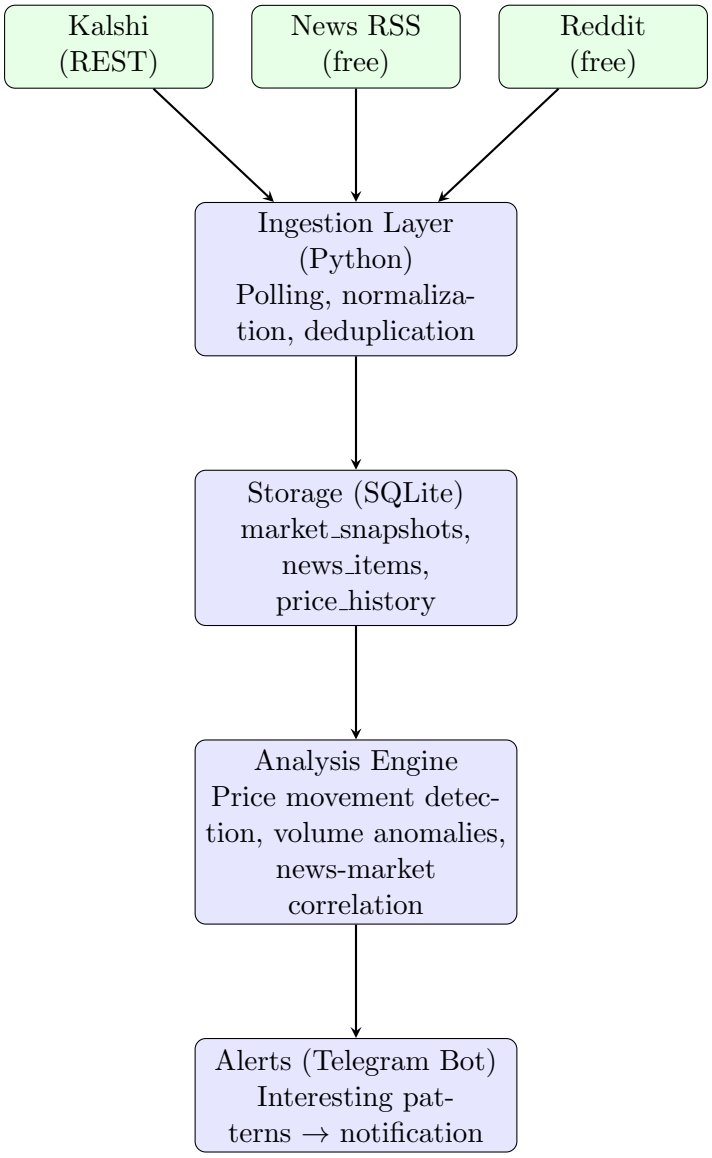
3 Proposed Architecture

The system follows a standard data pipeline architecture with four layers: ingestion, storage, analysis, and alerting.

System Components

- **Data Source:** Kalshi REST API (authenticated access available)
- **Ingestion Layer:** Python polling services with normalization and deduplication
- **Storage:** SQLite database for market snapshots, news items, and price history
- **Analysis Engine:** Price movement detection, news-market correlation, volume anomaly detection
- **Alerts:** Telegram bot for real-time notifications of interesting patterns

Data Flow



4 Technology Choices

Component	Technology	Justification
Backend	Python (FastAPI)	Familiar stack, excellent async support
Database	SQLite	Simple local storage, sufficient for prototype
Market API	Kalshi REST API	Documented API, authenticated access available
Signal Sources	RSS, Reddit API	Free tier access, reasonable rate limits
Alerts	Telegram Bot	Simple bot creation with BotFather, instant notifications
Hosting	Railway	Existing subscription, easy deployment

5 Two-Week Timeline

5.1 Week 1: Data Pipeline

- Set up Kalshi API integration with authentication
- Implement market list retrieval, current prices, and historical data collection
- Design and implement database schema for market snapshots
- Build basic polling loop with persistent storage
- Deploy initial data collection to Railway

5.2 Week 2: Signal Layer + Analysis

- Add news/RSS signal ingestion for 2–3 selected test markets
- Implement anomaly detection (price moves >X% without signal, signal without price response)
- Build Telegram alert bot for detected patterns
- Document initial findings and patterns observed

6 Research Questions

This exploration aims to gather empirical data on the following questions:

1. **Latency:** How quickly do Kalshi markets incorporate public news? Does this vary by market type (political vs. economic vs. weather)?
2. **Signal quality:** Which external sources (news RSS, Reddit, official announcements) correlate most strongly with subsequent price movements?
3. **Volume as signal:** Does order book depth or volume spike predict price movement direction or stability?
4. **Market type efficiency:** Are certain categories of markets (e.g., Fed decisions, elections, sports) more or less efficient than others?

7 Known Risks and Mitigations

Risk	Mitigation
API rate limits	Implement exponential backoff, cache aggressively, prioritize high-interest markets over broad coverage
Signal noise	Start with markets that have clear, identifiable catalysts (scheduled announcements, Fed decisions, election results)
Scope creep into trading	Explicit constraint: No automated trading in this phase. Analysis and alerts only.
Data source costs	Use only free tiers; RSS feeds as primary signal source; Reddit API free tier

8 Success Criteria

By the end of Week 2, the project will be considered successful if:

- Continuous data collection is running from Kalshi API
- At least one documented case study of signal-price relationship (either "signal preceded price" or "price moved without obvious signal")
- Working Telegram alert system delivering notifications for detected anomalies
- Written documentation of observed patterns and preliminary answers to research questions

9 Connection to Course Learning Outcomes

While this is a pre-project exploration before the main syllabus projects begin, it exercises several course learning outcomes:

- **Integrate external APIs and data sources:** Handling authentication, rate limiting, and data ingestion from the Kalshi platform
- **Deploy and document software systems:** Containerizing for Railway deployment, maintaining version control
- **Write technical proposals:** This document demonstrates clear problem statements, architecture specifications, and timeline planning