Prediction Market Edge Finder

Prediction Market Edge Finder is a professional-grade analytics dashboard that identifies statistical edges in Kalshi prediction markets for SPX, Nasdaq, BTC, and ETH. It combines real-time market data, AI-powered probability models (LightGBM), multi-source sentiment analysis, and a "Bloomberg Terminal" style interface.

⚡ Key Features

- **Real-Time Market Scanner**: Fetches and categorizes live Kalshi markets into Hourly, End of Day, and Range opportunities.
- **AI-Driven Probability**: LightGBM regressors for hourly (1-min data) and daily (1-hr data) predictions with auto-retraining on feature drift.
- **Kalshi Market Scanner**: Dedicated tab that scans all assets, calculates edge and Kelly sizing, and renders signal cards.
- **Sentiment Analysis**: Composite sentiment scoring from 3 free sources — Crypto Fear & Greed Index, VIX-derived sentiment, and price momentum — with averages display.
- **Opportunity Detection**: Edge calculation, moneyness filtering, and Alpha Picks highlighting.
- **Dark-mode "Bloomberg" aesthetic** with asset pills, integrated PnL simulator, and live market context.

🏗️ Architecture

Layer	Technology
Frontend	Streamlit
Modeling	LightGBM
Price Data	YFinance
Market Data	Kalshi API
Sentiment	alternative.me Fear & Greed API, VIX, price momentum
Config	YAML (`config/settings.yaml`)

📂 Project Structure

```
.
├── streamlit_app.py          # Main dashboard (2 tabs: Edge Finder +
Scanner)
├── config/
│   └── settings.yaml         # Centralized configuration
├── src/
│   ├── data_loader.py        # YFinance data fetching
│   ├── feature_engineering.py # Technical indicators (RSI, MACD, etc.)
│   ├── model.py              # LightGBM hourly model logic
│   ├── model_daily.py        # Daily model logic
│   ├── kalshi_feed.py        # Kalshi API integration
│   ├── market_scanner.py     # Market scanner (scan, signals, UI)
│   ├── sentiment.py          # Multi-source sentiment analysis
│   ├── signals.py            # Trading signal generation
│   ├── evaluation.py         # Model performance metrics
│   ├── utils.py              # Helper functions
│   └── azure_logger.py       # Azure Blob Storage logging
├── pages/
│   └── 1_Performance.py      # Performance analytics page
├── scripts/                  # Utility scripts
├── tests/                    # Test pipeline
├── model/                    # Saved .pkl models (gitignored)
└── CODEBASE_OVERVIEW.md      # Detailed codebase documentation
```

🚀 Setup & Installation

Prerequisites

- Python 3.9+

Installation

```
git clone <repository-url>
cd <repository-directory>
pip install -r requirements.txt
```

Environment Setup

Create a `.env` file in the root:

```
KALSHI_API_KEY=your_kalshi_api_key_here
# Optional: AZURE_CONNECTION_STRING=...
```

🖥️ Usage

```
streamlit run streamlit_app.py
```

- **Tab 1 — Kalshi Edge Finder**: Select asset → view opportunities → analyze edge
- **Tab 2 — Market Scanner**: Click "Scan Markets" → view signal cards with edge & Kelly sizing
- **Sentiment**: Expand the sentiment panel in either tab for composite scores and averages

⚠️ Disclaimer

This tool is for informational and educational purposes only. Prediction markets are high-risk. Trade at your own risk.