# **Comprehensive Guide to Stock Market Forecasting: Concepts, Data, Modeling, and Strategy**

This guide outlines a phased approach to building a sophisticated stock market forecasting system. It integrates market basics, data handling, feature engineering, advanced modeling techniques (Machine Learning/Deep Learning), evaluation, and strategic decision-making, incorporating technical, fundamental, and macroeconomic perspectives. The goal is to predict future stock prices, volatility, and risk, translating these forecasts into actionable Buy/Sell/Hold decisions for multiple companies.

## **🔁 End-to-End Project Flowchart (Conceptual Overview)**

[Raw Data Sources] → [Data Collection Scripts] → [Data Cleaning/Alignment] → [Feature Engineering] → [Modeling (ML/DL)] → [Evaluation] → [Trading Strategy Logic] → [Dashboard/Deployment/Backtesting]

## **🔀 ML Pipeline Diagram**

Raw Data → Preprocessing → Feature Engineering → Model Training → Model Validation → Signal Generation → Strategy Logic → Alerts/UI/API

## **🗂️ Team-Based Execution Plan (2.5 Weeks, 5 People)**

### **📅 Weekly Timeline with Milestones**

| **Week** | **Days** | **Milestones** |
| --- | --- | --- |
| Week 1 | 1–5 | ✅ Learn concepts, collect data, clean + merge datasets |
| Week 2 | 6–12 | ✅ Feature engineering, model training (ML/DL), implement strategy engine |
| Week 3 | 13–18 | ✅ Evaluate models, backtest strategy, build dashboard, final review & deploy |

### **👥 Team Role Assignment**

| Role | Assigned To | Responsibilities |
| --- | --- | --- |
| Project Manager | Member A | Coordination, task assignment, check-ins, GitHub repo maintainer |
| Data Engineer | Member B | Data collection (yfinance, FRED), cleaning, merging datasets |
| Feature Engineer | Member C | Build features (RSI, MACD, EPS growth, CPI lag), organize datasets |
| Model Developer | Member D | Train models (LSTM, XGB, Prophet), tune hyperparameters |
| Dashboard Developer | Member E | Streamlit dashboard, visualization, backtesting UI, alerts setup |

## **✅ Project Backlog (Detailed Task Breakdown)**

### **🧱 Week 1: Foundations & Data**

* ✅ Set up GitHub repo, clone locally (PM)
* ✅ Document team learning goals and glossary (All)
* ✅ Download OHLCV for 5+ tickers (Data Eng)
* ✅ Pull CPI, Interest Rates from FRED (Data Eng)
* ✅ Convert macro to daily and merge (Data Eng)
* ✅ Clean and align datasets (remove nulls, duplicates) (Data Eng)

### **🧠 Week 2: Features & Modeling**

* ✅ Calculate RSI, MACD, ATR (Feature Eng)
* ✅ Add EPS, P/E from FMP (Feature Eng)
* ✅ Normalize features (Feature Eng)
* ✅ Create lagged and rolling features (Feature Eng)
* ✅ Train ARIMA baseline model (Model Dev)
* ✅ Train XGBoost (Model Dev)
* ✅ Train LSTM with sequence input (Model Dev)
* ✅ Implement threshold logic and signal engine (Model Dev)

### **📊 Week 3: Evaluation, UI, Testing**

* ✅ Evaluate metrics (MAE, RMSE, Sharpe) (Model Dev)
* ✅ Build Streamlit dashboard (Dash Dev)
* ✅ Create watchlist panel (Dash Dev)
* ✅ Add Telegram/email alert triggers (Dash Dev)
* ✅ Backtest Buy/Sell logic (Dash Dev + Model Dev)
* ✅ Final team walkthrough (PM)
* ✅ Push full code with README to GitHub (PM)

## **📋 Kanban Board Template**

| To Do | In Progress | Testing/Review | Done |
| --- | --- | --- | --- |
| Download data | Clean stock prices | Validate CPI merge | Raw AAPL saved |
| RSI calc | Build feature df | Confirm RSI values | RSI verified |
| Train XGB | Optimize LSTM | Evaluate RMSE | LSTM done |
| Build dashboard | Add email alerts | Backtest signals | Strategy tested |

## **🧾 Full Requirements Checklist by Phase**

### **🔍 Phase 1 – Market Education**

### **📥 Phase 2 – Data Collection**

### **🧹 Phase 3 – Data Preprocessing**

### **🔧 Phase 4 – Feature Engineering**

### **🧠 Phase 5 – Modeling**

### **📊 Phase 6 – Evaluation**

### **🎯 Phase 7 – Strategy Logic**

### **🌐 Phase 8 – UI + Deployment**

# **🧭 Project Management Plan (2.5 Weeks, 5 People)**

## **🔧 Tools for Collaboration**

| **Tool** | **Purpose** |
| --- | --- |
| **GitHub / GitLab** | Version control, repo hosting |
| **Google Drive / Notion** | Docs, shared references |
| **Trello / Notion Board** | Task assignment, progress tracking |
| **Slack / Discord / WhatsApp** | Quick team communication |
| **Google Meet / Zoom** | Weekly team stand-ups & demos |

## **📆 Suggested Timeline (17-18 Days)**

### **🔹 Week 1: Foundations + Data + Prep**

| **Day** | **Milestone** |
| --- | --- |
| Day 1 | Kickoff + Team roles + Repo setup + Goals defined |
| Day 2–3 | Market & modeling theory learning (Crash course) |
| Day 4–5 | Data collection + cleaning + merging datasets |

### **🔹 Week 2: Features + Modeling + Strategy**

| **Day** | **Milestone** |
| --- | --- |
| Day 6–7 | Feature engineering (technical/fundamental/macro) |
| Day 8–10 | Train initial ML & DL models (RF, LSTM, Prophet) |
| Day 11–12 | Build strategy logic: Buy/Hold/Sell engine |

### **🔹 Week 3: Evaluation + Dashboard + Polish**

| **Day** | **Milestone** |
| --- | --- |
| Day 13–14 | Evaluate models + backtest strategy |
| Day 15 | Build Streamlit dashboard |
| Day 16 | Final testing + documentation + polish |
| Day 17–18 | Presentation prep + Final demo + GitHub push |

## **👥 Suggested Team Roles (with rotation if needed)**

| **Role** | **Responsibility** |
| --- | --- |
| **Project Manager (PM)** | Organize tasks, set deadlines, manage check-ins |
| **Data Wrangler** | Collect, clean, and align all data sources |
| **Feature Engineer** | Create technical indicators, fundamentals, macro features |
| **Model Developer** | Implement ML/DL models and evaluation logic |
| **Dashboard Dev** | Build the Streamlit app, visualizations, backtester |

🌀 Tip: Let everyone rotate or shadow other roles to **maximize learning**.

## **✅ Taskboard Example (Trello / Notion Kanban)**

| **To Do** | **In Progress** | **Review** | **Done** |
| --- | --- | --- | --- |
| Gather AAPL data | Merge CPI data | LSTM tuning review | Cleaned OHLCV CSV |
| Build RSI script | Normalize macro | Model metric table | SMA visual done |

## **📌 Key Practices for Success**

* 🔁 **Daily Stand-up (15 min)**: What I did, what I’m doing, blockers
* 🔂 **Mid-week checkpoints**: Demo progress to team
* 📤 **Push commits daily**: Use branches (e.g., model-dev, feature-eng)
* ✅ **Mini goals per day**: Keep progress visible
* 🎯 **Finish core model by Day 12**: Gives 5 days for polish