



Team Onboarding Guide: Stock Forecasting Project

Welcome to the StockSeer project — a collaborative capstone initiative where we'll use data science, machine learning, and financial knowledge to build a predictive and actionable stock market forecasting system.



Project Objective

To forecast the **future price, volatility, and risk** for multiple companies using technical indicators, macroeconomic data, and fundamental metrics — and to convert those predictions into intelligent **Buy/Hold/Sell trading signals**.

This project mimics a **real-world hedge fund workflow**, incorporating:

- Time series forecasting (LSTM, XGBoost, ARIMA, etc.)
 - Strategy engine logic (e.g., price vs. volatility scoring)
 - Model tracking & comparison with **MLflow**
 - Deployment-ready dashboard using **Streamlit**
-



Key Concepts to Understand

Financial & Technical Terms

Term	Meaning
OHLCV	Open, High, Low, Close, Volume (stock trading data)
RSI/MACD	Technical momentum indicators used to detect trends
P/E, EPS, ROE	Company valuation & profitability measures
CPI, Fed Rate, VIX	Macro indicators affecting all markets
Sharpe Ratio	Measures risk-adjusted return
Breakout/Hold/Support	Price action behaviors traders use to make decisions

Project Phases Summary

Phase 1: Setup + Exploration (Week 1)

- Define roles & goals
- Set up environments (Python, MLflow, Streamlit)
- Download and clean data (OHLCV, macro, fundamentals)
- Perform initial EDA and baseline modeling (ARIMA)

Phase 2: MVP Build (Week 2)

- Engineer features: RSI, lag, rolling stats, EPS growth, etc.
- Train & log ML/DL models (XGBoost, LSTM)
- Log runs with MLflow and analyze model results

Phase 3: Finalization (Week 3)

- Final model selection and tuning
 - Build dashboard + create strategy logic (Buy/Hold/Sell)
 - Backtest and document results
 - Prepare final presentation
-

Tools You'll Use

- **Python 3.9+:** Core programming language
 - **yfinance, fredapi:** For data collection
 - **pandas, numpy, sklearn:** Data wrangling
 - **keras, xgboost, statsmodels:** Modeling libraries
 - **MLflow:** Track parameters, metrics, and models
 - **Streamlit:** Build an interactive forecasting dashboard
-

How You Can Prepare

Learning Prep

- Watch tutorials on MLflow and Streamlit (1–2 hrs)
- Review EDA and modeling basics in pandas/sklearn
- Read about RSI, Bollinger Bands, and technical indicators
- Study one forecasting model (e.g., ARIMA or LSTM)

Attitude & Workflow

- Ask questions early and often
 - Communicate progress clearly
 - Commit to GitHub daily (even small changes)
 - Respect deadlines and demo days
-

Your Role & How to Contribute

Each person will lead a focus area (data, features, models, dashboard, or coordination), but everyone is encouraged to explore all parts of the pipeline. Use the Kanban board (project_tasks.kanban.md) to track your assignments.

Role	Contribution
PM (Julia)	Schedule, setup, repo maintenance, MLflow coordinator
Data Engineer (Paya)	Data collection & cleaning
Feature Engineer (Benassar)	RSI, lag, rolling features
Model Developer (Daniel)	Train/test models, track metrics
Dashboard & Docs (Kevin)	UI, visualizations, final presentation

Outcome

By the end of this project, you'll:

- Understand financial time series and ML applications in finance
- Have built an ML- and DL-based trading signal engine
- Gain experience using MLflow and Streamlit in a production-style workflow
- Have a deployable portfolio project hosted on GitHub

Let's build something real. 