Stock Market AI Analytics — Analysis Summary

Executive Summary

This document summarizes the recent work performed on the Stock Market AI Analytics dashboard during the development session. It captures key fixes, UX changes, data-loading behavior, and recommendations to include in a research paper.

Key Changes Applied

- Implemented robust sidebar filters: date-range handling, regime selector, model multiselect, and toggle to show model predictions. - Fixed `st.date_input` handling to accept both single-date and (start,end) tuples safely. - Repaired truncated and garbled lines in the code base that caused runtime/compile errors. - Removed the Model Comparison page per user request and updated navigation and quick-links accordingly. - Restored utility helpers `create_header()` and `create_status_metrics()` after removal introduced NameErrors. - Added an automatic loader (temporarily) to find model comparison summary files; later removed when page deleted.

Data and Environment

- Sample/stationary data is read from `stationary_data.csv` when available; fallback synthetic data used otherwise. - Prediction history is loaded from `data/cache/prediction_history.csv` if present. - Models are loaded from the `models/` folder; dummy placeholders are used when models are absent. - Development environment: Python virtualenv, Streamlit app served locally on port 8501.

Observations

- The dashboard compiles and runs locally after the fixes; Streamlit served successfully at http://localhost:8501. - Several deprecation warnings about Streamlit API (e.g., `use_container_width`) were observed; they are non-blocking but should be updated for future compatibility. - Model comparison functionality depended on external CSV/JSON artifacts; absence leads to empty-state messages. - Tests and some analysis scripts may still reference the removed Model Comparison page; update tests if needed.

Recommendations for Research Paper

1. Document the data provenance and preprocessing pipeline (`stationary_data.csv`, `preprocessing_pipeline.py`). 2. Include a short section describing UI changes and how filters affect analysis (date-range/regime/model selection). 3. Note the robustness fixes (date handling, missing-file fallbacks, helper restorations) as part of engineering rigor. 4. For model comparison results, specify required output schemas and include example CSV/JSON in the repository to reproduce figures. 5. Replace deprecated Streamlit APIs to future-proof reproducibility.

Appendix: Files Modified / Reviewed

- `app.py` — main entry, sidebar filters, header/status helpers, data/model loading. - `pages/04_model_comparison.py` — inspected and deleted per request. - `pages/01_home.py`, `temp_home.py` — quick-links updated to reflect page removal. - Other scripts reviewed: `comparison_report.py`, `model_evaluation.py`, `results_logger.py`.