Exploratory data analysis report

This report summarizes the dataset characteristics, preprocessing outcomes, engineered features, and key insights derived from your EDA code. It's structured for submission and aligns with your pipeline architecture and final app behavior.

Dataset overview

- **Source & schema:** Daily OHLCV and market cap per cryptocurrency, with columns normalized to lowercase and `marketcap` renamed to `market_cap`.
- **Time index:** `date` parsed to datetime; data sorted by `crypto_name, date` and deduplicated on that pair.
- **Types & ranges:** Numeric coercion applied to price/volume/cap; non-negative clipping for `volume` and `market_cap`.
- **Shape & coverage:**
    - **Rows/columns:** Report the final `df.shape` after cleaning.
    - **Assets:** List of unique `crypto_name` values.
    - **Date range:** Minimum and maximum `date`.
- **Missingness:** Count of missing values per column before feature engineering; note reductions after rolling features and target creation.

> Tip: Include a small table with counts per crypto (rows, start/end date) to show coverage consistency.

Preprocessing summary

- **Schema normalization:** Lowercased columns; removed `unnamed: 0` if present.
- **Type coercion:** Converted OHLCV and `market_cap` to numeric; invalid strings coerced to NaN.
- **Deduplication:** Dropped duplicate `(crypto_name, date)` pairs.
- **Non-negativity:** Clipped `volume` and `market_cap` at zero to avoid negative liquidity artifacts.

> Rationale: Ensures consistent, model-ready inputs and avoids downstream errors in volatility and liquidity computations.

Feature engineering

Returns and rolling volatility

- **Log return:**

log_rett=ln(close_t/(close_t−1))

- **Rolling volatility:**

vol_7d_t=std(log_rett−6:t),vol_14d_t=std(log_rett−13:t)

Garman–Klass volatility (daily proxy)

- **GK estimator (as implemented):**

GKt=(0.5 · (lnhightlowt)^2−2·(lnclosetopent)^2)^1/2

- **Note:** The classic GK includes an additional term with ln(close/open) and a coefficient; your simplified form is acceptable as a proxy but document the choice.

Liquidity and spreads

- **High–low spread:**

hl_spread_t= (high_t−low_t) /close_t

- **Turnover ratio:**

turnover_t= volume_t / market_cap_t

Calendar features

- **Day of week:** 0–6 (Mon–Sun)
- **Month:** 1–12

Target variable

- **Next-day GK volatility:**

target_vol_t=GK_t +1

After feature creation, rows with NaNs (from rolling windows and shifts) are dropped; features are saved to `data/processed/features.parquet`.

Visual analysis

Include the following plots (export as images and embed):

- **Closing price trends (per asset):**
  - **What to show:** Line plots of `close` for 3–5 representative assets.

    - **Insight:** Regime shifts, drawdowns, and recovery periods.
- **Volume trends (per asset):**
    - **What to show:** Line plots of `volume` for 3 assets.
    - **Insight:** Liquidity cycles; spikes around events.
- **Feature correlation matrix:**

What to show: Heatmap of

    - `['log_ret','vol_7d','vol_14d','garman_klass','hl_spread','turnover','target_vol']`.
    - **Insight:**
        - **Expected:** Positive correlation between `garman_klass` and rolling vol; `hl_spread` correlates with GK; `turnover` may inversely correlate with spreads for large caps.
- **Distribution plots (optional but valuable):**
    - **Histograms/KDEs:** `log_ret`, `garman_klass`, `vol_14d`.
    - **Insight:** Heavy tails in returns; skew in GK; heteroskedasticity.
- **Calendar effects (optional):**
    - **Boxplots:** GK by `day_of_week`.
    - **Insight:** Weekend/weekday differences; Monday spikes.

Key findings

- **Volatility structure:** Rolling vol and GK move together, confirming GK as a viable target proxy.
- **Liquidity–volatility link:** Higher `hl_spread` aligns with higher GK; `turnover` varies by asset class—small caps show higher volatility and unstable liquidity.
- **Data quality:** Initial missingness is addressed; rolling windows and target shift reduce usable rows—document final row count post-drop.
- **Seasonality:** Day-of-week effects may be present; quantify if notable.