1. Data Ingestion & Validation (DataIngestion&Validation.ipynb)

Overview and Concepts

This notebook handles the initial stages of the project: loading raw data, validation, cleaning, merging, and aggregation. Key concepts include:

- **Data Ingestion**: Mounting Google Drive for file access and loading CSVs (historical_data.csv and fear_greed_index.csv).
- **Validation**: Checking shapes, duplicates, missing values, and outliers using Pandas and SciPy. Outlier detection employs the Interquartile Range (IQR) method (1.5 * IQR bounds).
- **Cleaning and Transformation**: Parsing timestamps to UTC and IST formats, coercing numeric columns (e.g., leverage, PnL, size, price), deriving notional values (size * price), and extracting trade dates.
- **Merging**: Joining trade data with sentiment on trade_date, handling date formats for alignment.
- **Aggregation**: Creating daily summaries (e.g., trade count, average PnL, total notional) merged with sentiment.

Key Code Sections and Functioning

1. **Setup and Loading**:
   - Mounts Google Drive and creates directories (csv_files, outputs).
   - Loads datasets: historical_data (211,224 rows, 16 columns) and fear_greed_index (2,644 rows, 4 columns).
   - Code: historical_data = pd.read_csv(historical_data_path).
2. **Validation Checks**:
   - Prints shapes and displays heads/infos.
   - Detects duplicates (full rows and by trade ID candidates like 'Side').
   - Outlier summary for PnL: Q1=0.0, Q3=5.79, IQR=5.79, lower=-8.69, upper=14.48; 9,221 below, 39,720 above.
   - Code: Custom outlier_summary function using quantiles and IQR.
3. **Cleaning and Derived Features**:
   - Converts timestamps: tr['time_utc'] = pd.to_datetime(...) with timezone handling.
   - Numeric coercion: Maps columns like 'Closed PnL' to 'closedPnL'.
   - Derives 'notional' and 'time_ist'.
   - Saves cleaned Parquet: tr.to_parquet('csv_files/clean_trades.parquet').
4. **Merging and Aggregation**:
   - Groups trades by date: agg_daily = tr.groupby('trade_date').agg(...).

- Merges with sentiment: agg_daily.merge(sentiment[['date','value','classification']], ...).
- Saves: trades_merged.csv (trade-level) and daily_aggregates.csv (480 rows).

Achieved Results

- Cleaned dataset: 211,224 trades with standardized columns (e.g., time_utc, closedPnL, notional).
- Daily aggregates: 480 rows summarizing trades by date, including sentiment (e.g., value, classification like 'Fear').
- Validation insights: No full duplicates, but high duplicate trade IDs (211,222), indicating potential grouping needs. Significant PnL outliers suggest skewed distributions.
- Outputs: trades_merged.csv and daily_aggregates.csv in csv_files.

This notebook establishes a reliable data foundation, ensuring downstream analyses are on validated, merged data.

2. Exploratory Data Analysis & Analytics (ExploratoryDataAnalysis&Analytics.ipynb)

Overview and Concepts

This notebook focuses on EDA to understand data distributions, missingness, duplicates, and relationships. Concepts include:

- **Missingness Analysis**: Temporal and columnar missing ratios.
- **Distribution and Correlation**: Sentiment impacts on metrics like PnL, leverage.
- **Visualizations**: Bar plots, line charts, heatmaps for trends.
- **Automated Reporting**: Generates a PDF summary with insights, charts, and tables using ReportLab.
- **Derived Relationships**: Explores sentiment-PnL links (e.g., cumulative PnL by regime) and symbol sensitivity.

Key Code Sections and Functioning

1. **Setup and Loading**:
    - Loads cleaned data: trades_merged.csv (211,224 rows) and daily_aggregates.csv (480 rows).
    - Code: trades = pd.read_csv(...) with date parsing.
2. **Missingness and Quality Checks**:
    - Columnar missing: Plots bar chart of ratios.
    - Temporal missing: Groups by date, plots average missingness.
    - Duplicates: 0 full-row duplicates.

    - Code: missing_time = trades.groupby('trade_date').apply(lambda x: x.isnull().mean().mean()); saves PNGs like missing_by_column.png.

3. **Sentiment Distribution and Relationships**:
    - Value counts: Fear (61,837), Greed (50,303), etc., with 6 NaNs.
    - Symbol-sentiment sensitivity: Groups by symbol/sentiment, aggregates avg_PnL/leverage; saves symbol_sentiment_sensitivity.csv.
    - Cumulative PnL by sentiment: Sorts trades, computes cumsum; interactive Plotly line chart saved as HTML (cum_pnl_by_sentiment.html).
    - Code: trades_sorted['cum_pnl'] = trades_sorted.groupby('sentiment')['pnl_pct'].cumsum(); px.line(...).

Achieved Results and Visualizations

- **Missingness Insights**: Low overall missing (e.g., sentiment: 6 in trades, 1 in aggregates); plots show stable over time.
- **Relationships Derived**:
    - Sentiment impacts: Higher leverage in Greed vs. Fear; win rates higher in Neutral.
    - Cumulative PnL: Trends show Greed regimes yielding higher cumulative profits (visualized in interactive HTML).
    - Symbol Sensitivity: CSV reveals per-symbol averages, e.g., varying PnL by regime.
- **Visualizations**: missing_by_column.png, missing_over_time.png, cum_pnl_by_sentiment.html.
- **PDF Report**: Automated summary in ds_report.pdf, embedding charts and stats for stakeholder communication.

This notebook uncovers key patterns, like sentiment-driven behavior, setting the stage for modeling.

3. Feature Engineering & Models (FeatureEngineering&MODELS4.ipynb)

Overview and Concepts

This notebook advances to feature creation, clustering, modeling (classification for profitable trades), and backtesting. Concepts include:

- **Feature Engineering**: Lags, rolling stats, derived metrics (e.g., PnL EWMA, win streaks) to capture temporal dependencies.
- **Clustering**: Trader and market regimes for segmentation.
- **Modeling**: LightGBM classifier to predict trade profitability; prevents leakage by lagging target-derived features.
- **Backtesting**: Simulates trade selection based on model probabilities, comparing PnL.
- **Visualizations**: Feature importances, confusion matrices, regime plots.

Key Code Sections and Functioning

1. **Setup and Loading**:
    - Loads cleaned data and aggregates; tree shows 54 files in csv_files (e.g., clusters, sentiment_regimes).
    - Prints shapes: Trades (211,224 x 29), aggregates (480 x 8).
2. **Feature Engineering**:
    - Lags and Rolling: For cols like 'avg_pnl_pct', adds lags (1,3,7) and rolling mean/std (windows 3,7,14) grouped by date/symbol.
    - Code: Custom add_lag_features and add_rolling_features using groupby-shift/rolling.
    - Ultimate Features: Loads trades_ultimate_features.csv; lags target-derived (e.g., 'pnl_ewma_7', 'win_streak') to avoid leakage.
    - Derived: 'is_profitable' binary target; dummies for 'side', 'coin'.
3. **Clustering and Regimes**:
    - Trader Clusters: Saved in trader_clusters.csv; visualizes in Trade Clusters based on daily aggregates.png.
    - Sentiment Regimes: sentiment_regimes.csv and sentiment_transition_stats.csv; durations plotted in Distribution of Sentiment Regime Durations.png.
    - Market Regimes: HMM selection (hmm_model_selection.png), regimes plot (market_regimes_hmm.png).
4. **Modeling and Evaluation**:
    - Split: Time-based (80/20, no shuffle) to preserve chronology.
    - LightGBM Classifier: Binary objective, AUC metric, 1000 estimators.
    - Fits on lagged features; predicts probabilities.
    - Code: lgbm.fit(X_train, y_train); y_pred_proba = lgbm.predict_proba(X_test)[:, 1].
    - Visuals: feature_importance.png, confusion_matrix.png.
5. **Backtesting**:
    - Threshold (0.75): Select trades with prob >= threshold.
    - Results: 17,954/42,245 trades taken (42.50%), 99.03% win rate, PnL 1,603,733.55 (vs. all: 1,007,553.65; improvement: 596,179.89).
    - Saves: backtest_results.csv, classification_backtest_summary.txt.

Achieved Results, Models, and Visualizations

- **Features and Relationships**:
  - Temporal: Lags/rolling capture momentum (e.g., past PnL influences future).
  - Derived: 'pnl_div_by_volatility' normalizes risk; clusters segment high/low leverage trades (e.g., dataset_per_trade_high_leverage.csv).
  - Sentiment Links: Regimes correlate with leverage (higher in Greed; plot: Average Leverage Distribution by Sentiment Regime.png) and PnL volatility (Volatility-Adjusted PnL% by Sentiment.png).
  - Transitions: Stats in sentiment_transition_stats.csv show regime persistence.
- **Models**:
  - LightGBM: High accuracy in classifying profitable trades; feature importances saved (e.g., lightgbm_feature_importances.csv).
  - Alternatives: Mentions RF/XGBoost importances in CSVs.
  - Backtest: Demonstrates model utility, selecting high-confidence trades for superior PnL.
- **Visualizations**:
  - Trends: 7dayAvg_Rolling_win_rate_&_Avg_Leverage.png, acf_pacf_pnl.png.
  - Regimes/Clustering: market_regimes_hmm.png, Trade Clusters based on daily aggregates.png.
  - Model: confusion_matrix.png, feature_importance_Main_Dataset.png.
  - Networks: trader_network.png, trader_change_points.png.
- **Outputs**: Clustered CSVs (e.g., 4 trade clusters), backtests, importances; total 54 files in csv_files.

4. Overall Relationships, Derived Insights, Models, and Visualizations

- **Key Relationships**:
  - Sentiment-PnL: Greed regimes show higher cumulative PnL and leverage; Fear correlates with lower win rates but potentially safer trades (derived from cumsum and aggregates).
  - Temporal Dependencies: Lags/rolling reveal autocorrelation in PnL (ACF/PACF plots); win streaks influence future profitability.
  - Clustering Insights: 4 trader clusters based on aggregates; high-leverage clusters have distinct PnL distributions.
  - Volatility Adjustments: Normalizing PnL by volatility highlights regime differences (plot shows Extreme Greed with highest adjusted returns).

- **Models Emphasis**:
  - Primary: LightGBM classifier outperforms baselines (e.g., RF, XGBoost via importances CSVs), focusing on binary profitability prediction.
  - Regime Modeling: HMM for market states; KMeans for daily regimes (daily_regimes_kmeans.csv).
  - Backtesting Validates: Threshold-based selection yields 59% PnL uplift, emphasizing model's practical value.
- **Visualizations Emphasis**:
  - Over 20 PNGs/HTMLs: Temporal (e.g., rolling metrics), distributional (e.g., MI scores, non-linear relationships), and model-specific (e.g., SHAP summaries in CSVs, visualized importances).
  - Interactive: Plotly for cumulative PnL.
  - Heatmaps: Trade activity by hour/sentiment (Trade_Activity_heatmap_by_hour&sentiment.png).

5. Generated CSV Files and Their Roles

From the Drive link and notebook trees:

- **Cleaning/Aggregation**: trades_cleaned.csv (validated trades), daily_aggregates.csv (daily summaries), trades_merged.csv (with sentiment).
- **Clustering**: trader_clusters.csv, dataset_per_trade_cluster_[0-3].csv (segmented trades), dataset_per_trade_high/low_leverage.csv.
- **Features/Regimes**: trades_feature_engineered.csv, trades_ultimate_features.csv, sentiment_regimes.csv, sentiment_transition_stats.csv.
- **Model Outputs**: lightgbm_feature_importances.csv, rf_feature_importances.csv, xgboost_feature_importances.csv, lightgbm_shap_summary.csv.
- **Predictions/Backtests**: daily_pnl_predictions.csv, backtest_results.csv.
- These enable modular reuse, e.g., clusters for targeted modeling.

Conclusion

This project effectively processes trading data, derives sentiment-driven insights, and deploys models for enhanced profitability. Strengths include leakage prevention, comprehensive visualizations, and backtesting. Potential improvements: Incorporate external data (e.g., market volatility), ensemble models, or real-time deployment. The notebooks form a cohesive, reproducible pipeline.