Contents

1 Asset Screening System

1.1 Complete Architecture Overview:

Key code file: `hqm_list.py`
Purpose: Intelligent asset filtering using proprietary trend, momentum and quality scoring.

The `hqm_list.py` script orchestrates a proprietary "Hedged Quality Momentum" (HQM), as an asset screening system that systematically evaluates crucial characteristics across nine distinct asset classes. Acting as a batch processing controller, it sequentially executes specialized analysis scripts for indices, equities, commodities, bonds, real estate, forex, cryptocurrencies, alternatives and futures markets.

At its most basic core, the system implements a multi-dimensional momentum scoring algorithm that calculates percentile rankings across eight key factors: closing prices, highs, lows, trading volume, and returns over 1-month, 3-month, 6-month, and 12-month periods. Each asset receives an HQM score representing the average percentile across these momentum indicators, with scores above 70 or below 30 triggering high-conviction filtering. It elaborates and synthesizes the investment approaches of **Gary Antonacci's Dual Momentum** and **Joe Terranova's Quality Momentum**, not as a wholesale investment strategy but as a filter for assets, a warrant that allows us to justify the "bet" on those assets for more computationally complex, costly and time consuming analyses.

The "Hedged" quality of the filter also comes from using further fundamental (William O'Neil's CAN SLIM and Benjamin Graham's value investing insights) and quantitative (Machine Learning models) "checks" of the data, which further filter out those remaining stocks which are less likely to do well.

The underlying `main_hqm.py` module handles part the technical implementation, downloading approximately 15 months of historical data via Yahoo Finance and applying weighted linear regression for 5-day price forecasts. This corresponds to an adjusted Gary Antonacci's Dual Momentum approximation.

`asset_quality.py` takes care of the methodologies discussed by Terranova in his Quality Momentum approach.

The system extends beyond this core functionality through multiple sophisticated analysis layers:

1.1.1 Advanced Quantitative Analysis (`asset_quant.py`)

A comprehensive ensemble modeling framework that integrates multiple machine learning algorithms including Linear Regression, XGBoost, ARIMA, and GARCH models to generate robust price forecasts with improved accuracy and reliability metrics. This module features Monte Carlo simulation for probabilistic forecasting, trade opportunity scoring, and comprehensive technical indicators.

1.1.2 CAN SLIM Fundamental Analysis (`asset_canslim.py`)

Implements William O'Neil's CAN SLIM methodology, analyzing Current quarterly earnings growth (C), Annual earnings growth (A), Institutional ownership patterns (I), Relative strength vs S&P 500 (S), Sector/market direction (L), and Market direction (M). Includes sophisticated volume pattern analysis for supply/demand assessment and ETF/Index detection capabilities.

1.1.3 Graham Valuation Analysis (`asset_graham.py`)

Applies Benjamin Graham's value investing principles through enhanced EPS estimates, industry peer analysis, risk-adjusted discount rates, historical valuation context, and advanced DCF projections. Provides comprehensive fundamental valuation metrics for identifying undervalued assets with strong intrinsic value characteristics.

This multi-layered ensemble approach calculates weighted averages of individual model predictions, with the framework incorporating comprehensive reliability metrics including RMSE, MAE, R^2 scores, directional accuracy, and trend strength analysis to validate forecast confidence.

Output management includes automated CSV generation with sector-specific results, benchmark comparisons against sector averages, and detailed execution summaries tracking success rates and processing times. This framework enables systematic identification of momentum anomalies across diverse asset

classes, providing quantitative signals for investment decision-making, but in our case it is merely as a prerequisite for the assets to qualify for a complete analysis.

To sum it up, this is essentially a **comprehensive multi-factor asset screening framework** that combines quantitative momentum analysis, machine learning ensemble forecasting, technical analysis, CAN SLIM fundamental screening, and Graham value investing principles to systematically scan multiple asset classes and identify statistically significant investment opportunities with multiple layers of validation.

1.2 Five-Stage Analysis Pipeline

The system has evolved into a sophisticated five-stage analysis pipeline that provides institutional-grade asset screening and investment decision support:

1.3 Stage 1: Core HQM Foundation Layer

1.3.1 1.1 Sequential Batch Processing Orchestrator (`hqm_list.py`)

This script serves as the **master controller** that: - **Sequentially executes** 9 specialized HQM scripts (not parallel processing) - Each script targets a different asset class: - `hqm_index.py` - Stock market indices - `hqm_stocks.py` - Individual equities (~100 major stocks) - `hqm_comm.py` - Commodities - `hqm_bonds.py` - Bond markets - `hqm_reals.py` - Real Estate - `hqm_forex.py` - Currency pairs - `hqm_crypto.py` - Cryptocurrencies (~80 assets) - `hqm_alts.py` - Alternative investments - `hqm_fut.py` - Futures contracts

There is also hqm_market.py, meant to be a way to assess the current behavior of the key (most influential) assets of the whole market. ### **1.2 Core HQM Analysis Engine (`main_hqm.py`)**

The **foundation algorithm** that implements:

```
def download_historical_data(symbol):
    # Downloads ~15 months of historical data via Yahoo Finance
    end_date = datetime.now()
    start_date = end_date - timedelta(days=456)
    data = yf.download(symbol, start=start_date, end=end_date)
```

Data Pipeline:

```
def calculate_returns(data):
    # Calculates returns across multiple timeframes
    min_periods = {
        '1m_return': 20,    # ~1 month
        '3m_return': 60,    # ~3 months
        '6m_return': 125,   # ~6 months
        '1y_return': 252    # ~1 year
    }
```

Multi-Timeframe Momentum Analysis:

```
def calculate_momentum_percentile(data):
    # Analyzes 8 momentum factors:
    momentum_factors = [
```

```
        'Close', 'High', 'Low', 'Volume',  # Price/volume data
        '1m_return', '3m_return', '6m_return', '1y_return'  # Time-based returns
    ]

    # Converts each factor to percentile ranking (0-100)
    # Higher percentile = stronger momentum
```

Percentile-Based Scoring:

```
def forecast_price(data, forecast_days=5):
    # Implements weighted linear regression forecasting
    # Uses exponential weighting (recent data more important)
    # Includes comprehensive reliability metrics
```

Price Forecasting with Reliability:

```
# Final HQM score = average percentile across all 8 momentum factors
hqm_score = sum(momentum_percentiles.values()) / len(momentum_percentiles)
```

HQM Score Calculation:

1.4 Stage 2: Advanced Quantitative Ensemble Analysis (`asset_quant.py`)

This **revolutionary layer** introduces **institutional-grade machine learning forecasting** that transforms the basic momentum foundation into a sophisticated predictive analytics engine:

1.4.1 2.1 Ensemble Forecasting System

Multi-Algorithm Integration:

- **Linear Regression**: Statistical baseline for trend analysis
- **XGBoost**: Gradient boosting for complex pattern recognition
- **ARIMA**: Autoregressive integrated moving average for time series modeling
- **GARCH**: Generalized autoregressive conditional heteroskedasticity for volatility forecasting

```
def _add_moving_averages(self):
    # Comprehensive moving average suite
    self.data['SMA_20'] = self.data['Close'].rolling(window=20).mean()
    self.data['SMA_50'] = self.data['Close'].rolling(window=50).mean()
    self.data['SMA_200'] = self.data['Close'].rolling(window=200).mean()
    self.data['EMA_12'] = self.data['Close'].ewm(span=12).mean()
    self.data['EMA_26'] = self.data['Close'].ewm(span=26).mean()

def _add_momentum_indicators(self):
    # MACD with signal line and histogram
    self.data['MACD'] = self.data['EMA_12'] - self.data['EMA_26']
    self.data['MACD_Signal'] = self.data['MACD'].ewm(span=9).mean()
    self.data['MACD_Histogram'] = self.data['MACD'] - self.data['MACD_Signal']
```

```
    # RSI (Relative Strength Index)
    # Bollinger Bands
    # Additional momentum indicators
```

Advanced Technical Indicators Engine:

```
def run_monte_carlo_simulation(self, num_runs=30, random_seed=None, bias_score=5
    # Advanced Monte Carlo simulation with directional bias
    # Noise injection for market uncertainty modeling
    # Confidence interval calculation (80% and 95%)
    # Statistical distribution analysis
    # Bias factor application for directional forecasting
```

Monte Carlo Probabilistic Forecasting:

```
class TradeOpportunityScorer:
    # Sophisticated scoring algorithm considering:
    # - Direction strength and consistency analysis
    # - Confidence interval width evaluation
    # - Risk-adjusted return calculations
    # - Signal strength multiplier application
    # - Final trade opportunity score (0-100) with interpretation
```

Trade Opportunity Scoring System:

1.4.2 2.2 Performance Validation & Ensemble Weights

The system automatically calculates optimal ensemble weights based on individual model performance, using inverse MSE weighting to ensure the most accurate models have greater influence in the final forecast.

1.5 Stage 3: Technical Analysis Integration (`thumb.py`)

This component **integrates multiple technical analysis systems** (dubbed "rules-of-thumb" originally, as they were meant as heuristic checks to speed up short term trading at a glance):

Multi-System Technical Analysis:

- **Day Analysis** (`day.py`): Short-term technical signals and intraday patterns
- **Simplified Technical** (`simptlest.py`): Core technical indicators (RSI, MACD, Bollinger Bands)
- **HQM Integration** (`hqm_short.py`): Momentum-based scoring on shorter timeframes
- **Sentiment Analysis** (`sen_text.py`): Market sentiment indicators
- **Advanced Technical** (`atr2.py`): Volatility and trend analysis with ATR

```
# Weighted scoring across multiple timeframes and systems
weighted_scores = []
if day_score_value is not None:
```

```
    weighted_scores.extend([day_score_value, day_score_value, day_score_value])
if simple_avg_value is not None:
    weighted_scores.append(simple_avg_value)
if hqm_score_value is not None:
    weighted_scores.append(hqm_score_value)
if sentiment_score_value is not None:
    weighted_scores.append(sentiment_score_value)
if variability_value is not None:
    weighted_scores.extend([moovy_groovy_value, moovy_groovy_value, variability_

avg_analysis_score = sum(weighted_scores) / len(weighted_scores)
```

Composite "Thumb" Score Calculation:

1.6 Stage 4: Fundamental Analysis Layer

This **critical layer** applies **time-tested fundamental investment methodologies** to validate quantitative signals with intrinsic value analysis:

1.6.1 4.1 CAN SLIM Fundamental Analysis (`asset_canslim.py`)

Implements William O'Neil's legendary CAN SLIM methodology:

C - Current Quarterly Earnings Growth:

- Analyzes most recent quarter-over-quarter earnings growth
- Compares year-over-year performance
- Flags accelerating earnings trends

A - Annual Earnings Growth Rate:

- Calculates compound annual growth rate (CAGR) over multiple years
- Identifies sustainable growth patterns
- Penalizes inconsistent earnings trajectories

I - Institutional Ownership Patterns:

- Analyzes institutional holdings percentage
- Identifies accumulation vs distribution patterns
- Flags optimal ownership ranges (typically 10-60%)

S - Relative Strength vs S&P 500:

- Compares 3-month performance vs broad market
- Identifies leadership characteristics
- Flags outperforming assets

L - Sector/Market Direction:

- Analyzes sector-specific trends and momentum
- Compares asset performance vs sector ETF benchmarks
- Identifies sector rotation opportunities

M - Market Direction:

- Assesses overall market trend strength
- Analyzes volume patterns for institutional commitment
- Provides market timing context

```
def analyze_volume_patterns(symbol, stock_data=None):
    # Advanced volume analysis for supply/demand assessment
    # Volume trend ratios and consistency metrics
    # Breakout volume detection
    # Volume spike frequency analysis
    # Price-volume correlation studies
```

Volume Pattern Analysis:

1.6.2 4.2 Graham Valuation Analysis (`asset_graham.py`)

Applies Benjamin Graham's value investing principles:

```
def get_enhanced_eps_estimates(symbol):
    # Analyst consensus estimates
    # Forward EPS projections
    # Historical EPS growth calculation
    # EPS quality scoring system
```

Enhanced EPS Analysis:

```
def get_industry_peer_analysis(symbol, sector):
    # Sector-specific peer group identification
    # Relative valuation metrics
    # Percentile ranking calculations
    # Competitive positioning analysis
```

Industry Peer Analysis:

```
def get_risk_adjusted_discount_rate(symbol, base_rate=0.10):
    # Company-specific risk assessment
    # Industry risk premium calculation
    # Dynamic discount rate determination
    # Risk-adjusted present value calculations
```

Risk-Adjusted Valuation:

```
def get_advanced_dcf_projections(symbol, fcf, growth_scenarios=None):
    # Multiple growth scenario modeling
    # Terminal value calculations
    # Sensitivity analysis
    # Probabilistic DCF valuation
```

Advanced DCF Projections:

Graham Valuation Metrics:

- **Graham Number**: `sqrt(22.5 * EPS * Book Value per Share)`
- **Margin of Safety**: Current Price vs Intrinsic Value
- **P/E Ratio Analysis**: Historical and forward P/E metrics
- **Debt-to-Equity Analysis**: Financial leverage assessment
- **Dividend Yield Analysis**: Income generation potential

1.7 Stage 5: Intelligent Filtering & Decision Layer (`Filter_real.py`)

Filter_real.py is the filter corresponding to the real estate assets, specifically. We are using it as the sample here, but each of the 9 asset categories gets one file with the same corresponding logic and structure; the only true variation between them is the specific sector or group of assets being screened.

1.7.1 5.1 Comprehensive Multi-Factor Analysis Integration

This **orchestrates** the complete analysis pipeline, now incorporating all five analysis stages:

```
# Reads HQM-filtered real estate symbols
csv_path = "/Users/davidsanchomarco/miniconda3/envs/analysis/Analizer-Predictor/
```

Data Source Integration:

```
# Extracts bias from HQM sector average and moderates extremes
hqm_score = sector_row['hqm_score'].iloc[0]
raw_bias = int(float(hqm_score))  # First 2 digits
self.bias_score = int((raw_bias + 50) / 2)  # Average with 50 to moderate
```

Bias Moderation System:

```
# Now runs comprehensive five-stage analysis for each symbol:
# 1. HQM momentum analysis (Stage 1)
# 2. Quantitative ensemble forecasting (Stage 2 - asset_quant.py)
# 3. Technical analysis integration (Stage 3 - thumb.py)
# 4. Fundamental analysis (Stage 4 - asset_canslim.py + asset_graham.py)
# 5. Intelligent filtering (Stage 5 - this layer)
```

Five-Stage Analysis Integration:

1.7.2 5.2 Advanced Signal Processing & Filtering

```
# Comprehensive scoring across all five analysis stages
# HQM Score (0-100): Momentum foundation
# Quant Score (0-100): ML ensemble forecasting
# Tech Score (0-100): Technical analysis composite
# CAN SLIM Score (0-100): Fundamental growth analysis
```

```
# Graham Score (0-100): Value investing metrics

# Weighted average calculation across all factors
if all(scores >= 0 for scores in [hqm_score, quant_score, tech_score, canslim_sc
    average_score = round((
        hqm_score * 0.15 +          # Foundation momentum
        quant_score * 0.25 +        # ML forecasting
        tech_score * 0.20 +         # Technical analysis
        canslim_score * 0.25 +      # Growth fundamentals
        graham_score * 0.15         # Value fundamentals
    ), 2)
```

Multi-Stage Score Integration:

```
# Advanced filtering considering consensus across multiple methodologies
filtered_data = self.data[
    # Strong consensus signals (all methodologies agree)
    ((self.data['Average Score'] < 35) & (self.data['Consensus_Strength'] >= 80)
    ((self.data['Average Score'] > 70) & (self.data['Consensus_Strength'] >= 80)
    # High-conviction single methodology signals
    (self.data['HQM_Score'] < 30) | (self.data['HQM_Score'] > 70) |  # Momentum
    (self.data['CAN_SLIM_Score'] > 75) |  # Strong fundamental signals
    (self.data['Graham_Score'] > 75) |    # Strong value signals
    (self.data['Average Score'] == -1)    # Failed analyses (preserve)
]
```

Intelligent Multi-Factor Signal Filtering: The enhanced filtering system now considers consensus strength across multiple methodologies, giving preference to signals that receive validation from multiple analytical approaches while still allowing high-conviction signals from individual methodologies to pass through.

1.7.3 5.3 Multi-Format Output System

```
# Saves comprehensive analysis results
output_path = self.csv_path.with_stem(f"{self.csv_path.stem}_comprehensive_analy
filtered_data.to_csv(output_path, index=False)

unfiltered_path = self.csv_path.with_stem(f"{self.csv_path.stem}_comprehensive_a
self.data.to_csv(unfiltered_path, index=False)

# Additional specialized outputs
consensus_path = self.csv_path.with_stem(f"{self.csv_path.stem}_consensus_signal
consensus_data.to_csv(consensus_path, index=False)
```

Enhanced CSV Generation:

1.8 Complete System Architecture

1.8.1 HQM Asset Screening System (2025 Version - Five Stage)

Core Architecture:

```
graph TD
    A[HQM Foundation] --> B[Quantitative Ensemble]
    B --> C[Technical Analysis]
    C --> D[Fundamental Analysis]
    D --> E[Intelligent Filtering]

    A --> F[Multi-Asset Momentum]
    B --> G[ML Ensemble Forecasting]
    C --> H[Technical Validation]
    D --> I[CAN SLIM + Graham]
    E --> J[Consensus Signals]

    style A fill:#e1f5fe
    style B fill:#f3e5f5
    style C fill:#e8f5e8
    style D fill:#fff3e0
    style E fill:#ffebee
```

Five-Stage Analysis Pipeline:

Stage	Component	Methodology	Output
1	**HQM Foundation**	Multi-asset momentum screening	Momentum scores (0-100)
2	**Quantitative Ensemble**	ML forecasting + Monte Carlo	Confidence intervals & trade opportunities
3	**Technical Analysis**	Multi-system composite	Technical validation scores
4	**Fundamental Analysis**	CAN SLIM + Graham valuation	Growth & value assessment
5	**Intelligent Filtering**	Consensus-based processing	High-conviction signals

Key Modules by Stage:

Stage	Primary Modules	Secondary Modules
Foundation	`hqm_list.py`, `hqm.py`	9 asset class scripts
Quantitative	`asset_quant.py`	Monte Carlo simulation
Technical	`thumb.py`	`day.py`, `simptlest.py`
Fundamental	`asset_canslim.py`, `asset_graham.py`	Industry analysis, DCF models
Decision	`Filter_real.py`	9 asset-specific filters

Comprehensive Output Results:

Score Type	Range	Description
HQM Scores	0-100	Momentum strength across 8 factors/timeframes
Quantitative Scores	0-100	ML ensemble forecasting with confidence intervals
Technical Scores	0-100	Multi-system technical analysis composite
CAN SLIM Scores	0-100	Growth investing fundamental analysis (C-A-N-S-L-I-M)
Graham Scores	0-100	Value investing fundamental analysis
Consensus Scores	0-100	Multi-methodology agreement strength
Trade Opportunity Scores	0-100	Final investment decision metric

Signal Generation Framework:

- **Consensus Signals**: Require >80% agreement across methodologies
- **High-Conviction Signals**: Strong individual methodology scores
- **Filtered Outputs**: Multiple formats (filtered, unfiltered, consensus-specific)
- **Risk Management**: Confidence intervals and bias moderation

---

1.9 Features Implemented

1.9.1 Five-Layer Analysis Integration:

- **Foundation Layer**: HQM momentum screening across 9 asset classes with 8-factor analysis
- **Quantitative Layer**: Advanced ML ensemble forecasting with Monte Carlo simulation
- **Technical Layer**: Multi-system technical analysis integration
- **Fundamental Layer**: Dual methodology analysis (CAN SLIM growth + Graham value investing)
- **Decision Layer**: Intelligent multi-factor signal processing and consensus filtering

1.9.2 Comprehensive Scoring System:

- **HQM Scores (0-100)**: Percentile-based momentum across 8 factors and timeframes
- **Quantitative Scores (0-100)**: ML ensemble forecasting with confidence intervals
- **Technical Scores (0-100)**: Weighted composite of multiple technical analysis systems
- **CAN SLIM Scores (0-100)**: Growth investing fundamentals (C-A-N-S-L-I-M analysis)
- **Graham Scores (0-100)**: Value investing fundamentals (intrinsic value, margin of safety)
- **Consensus Scores (0-100)**: Multi-methodology agreement strength measurement
- **Trade Opportunity Scores (0-100)**: Final investment decision metric with interpretation

1.9.3 Advanced Machine Learning Capabilities:

- **Ensemble Forecasting**: Linear Regression, XGBoost, ARIMA, GARCH integration
- **Monte Carlo Simulation**: Probabilistic forecasting with directional bias application
- **Trade Opportunity Scoring**: Sophisticated scoring considering direction, confidence, and risk
- **Performance-Based Weighting**: Automatic ensemble weight optimization
- **Confidence Interval Analysis**: 80% and 95% probability distributions

1.9.4 CAN SLIM Methodology Implementation:

- **Current Earnings Growth (C)**: Quarter-over-quarter and year-over-year analysis

- **Annual Earnings Growth (A)**: Compound annual growth rate calculations
- **Institutional Ownership (I)**: Accumulation vs distribution pattern analysis
- **Relative Strength (S)**: 3-month performance vs S&P 500 benchmark
- **Sector Leadership (L)**: Industry group analysis and sector rotation
- **Market Direction (M)**: Overall market trend and volume confirmation

1.9.5 Graham Value Investing Framework:

- **Enhanced EPS Analysis**: Analyst consensus and growth projections
- **Industry Peer Comparison**: Relative valuation and percentile rankings
- **Risk-Adjusted Discount Rates**: Company-specific cost of capital calculations
- **Advanced DCF Modeling**: Multi-scenario terminal value projections
- **Graham Number Calculation**: Intrinsic value estimation formula
- **Margin of Safety Analysis**: Conservative valuation with safety buffers

1.9.6 Intelligent Multi-Factor Signal Processing:

- **Consensus Strength Analysis**: Cross-validation across multiple methodologies
- **Bias Moderation System**: Prevents extreme directional bias
- **Multi-Threshold Filtering**: Different criteria for different signal types
- **High-Conviction Signal Preservation**: Maintains strong signals from individual methodologies
- **Failed Analysis Tracking**: Preserves records of unsuccessful analyses for review

1.9.7 Advanced Output Management:

- **Multi-Format CSV Generation**: Filtered, unfiltered, and consensus-specific outputs
- **Comprehensive Metadata**: Analysis timestamps, bias applied, methodology versions
- **Performance Metrics Export**: Model accuracy, confidence intervals, and validation statistics
- **Sector-Specific Benchmarking**: Performance vs sector ETF comparisons

1.9.8 Enterprise-Grade Error Handling:

- **Graceful Degradation**: System continues operating with partial component failures
- **Data Quality Validation**: Robust checking for data availability and integrity
- **Package Dependency Management**: Automatic fallback for missing optional packages
- **Comprehensive Logging**: Detailed execution tracking and debugging information

---

1.10 Real-World Trading Application

This system is designed to provide **institutional-grade analysis** for professional investment decision-making through a comprehensive five-stage methodology:

1.10.1 Advanced Signal Generation:

- **Consensus Bullish Signals**: Average Score > 70 with Consensus Strength > 80
- **Consensus Bearish Signals**: Average Score < 35 with Consensus Strength > 80
- **High-Conviction Individual Signals**: Strong signals from any single methodology
- **CAN SLIM Breakouts**: Scores > 75 indicating exceptional growth characteristics
- **Graham Value Opportunities**: Scores > 75 indicating significant margin of safety

1.10.2 Multi-Methodology Validation Framework:

- **HQM Momentum Validation**: Confirms directional strength across multiple timeframes
- **ML Ensemble Forecasting**: Provides probabilistic price predictions with confidence intervals
- **Technical Analysis Confirmation**: Multi-system validation of timing and entry points
- **CAN SLIM Growth Analysis**: Validates earnings acceleration and institutional accumulation
- **Graham Value Assessment**: Confirms intrinsic value and margin of safety requirements

1.10.3 Enterprise Risk Management:

- **Monte Carlo Confidence Intervals**: 80% and 95% probability distributions for price forecasts
- **Bias Moderation System**: Prevents overconfidence in extreme market conditions
- **Multi-Factor Consensus Filtering**: Requires agreement across methodologies for high-conviction signals
- **Sector Rotation Analysis**: Identifies optimal market timing based on sector leadership
- **Volume Pattern Confirmation**: Validates institutional commitment through supply/demand analysis

1.10.4 Portfolio Construction Applications:

- **Asset Allocation**: Multi-asset class screening across equities, bonds, commodities, forex, crypto
- **Sector Rotation**: Dynamic positioning based on CAN SLIM sector leadership analysis
- **Risk Parity**: Balanced exposure using Graham's risk-adjusted valuation framework
- **Long/Short Strategies**: Enhanced short opportunities with multi-factor validation
- **Hedge Fund Applications**: Institutional-grade analysis for alternative investment strategies

---

1.11 Practical Implementation Results

Each complete five-stage analysis run produces comprehensive outputs designed for different investment decision-making needs:

1.11.1 1. Multi-Asset Momentum Foundation (Stage 1)

- HQM-ranked assets across all 9 major asset classes (indices, stocks, commodities, bonds, real estate, forex, crypto, alternatives, futures)
- 8-factor momentum analysis with percentile rankings
- Sector-specific benchmarking and relative strength assessment

1.11.2 2. Advanced Quantitative Forecasting (Stage 2)

- ML ensemble predictions using Linear Regression, XGBoost, ARIMA, and GARCH
- Monte Carlo simulation results with 80% and 95% confidence intervals
- Trade opportunity scores with directional bias application
- Performance-based ensemble weighting for optimal accuracy

1.11.3 3. Technical Analysis Integration (Stage 3)

- Multi-system technical composite scores from 5+ analysis systems
- Short-term day analysis, core technical indicators, HQM integration
- Sentiment analysis and advanced volatility-based trend analysis
- Weighted scoring system for comprehensive technical validation

1.11.4 4. Fundamental Analysis Deep Dive (Stage 4)

- **CAN SLIM Analysis**: Complete C-A-N-S-L-I-M assessment with institutional accumulation patterns
- **Graham Valuation**: Intrinsic value calculations, margin of safety analysis, DCF projections
- **Industry Peer Comparisons**: Relative valuation and competitive positioning
- **Risk-Adjusted Metrics**: Company-specific discount rates and terminal value scenarios

1.11.5 5. Intelligent Decision Layer (Stage 5)

- Consensus strength analysis across all methodologies
- Multi-threshold filtering system for different signal types
- Weighted scoring system balancing all five analysis stages
- Enhanced output formats: filtered, unfiltered, and consensus-specific results

1.11.6 Comprehensive Output Formats:

- **Primary Analysis CSVs**: Complete scoring matrix with all methodology results
- **Filtered Signal Universe**: Only high-conviction opportunities meeting consensus criteria
- **Consensus Analysis Files**: Signals validated across multiple methodologies
- **Performance Metrics Export**: Model accuracy statistics and validation results
- **Sector Benchmarking Reports**: Performance vs sector ETF comparisons

This represents a **complete institutional-grade asset screening system** that combines traditional investment methodologies (CAN SLIM, Graham) with modern quantitative techniques (ML ensemble forecasting, Monte Carlo simulation), providing both breadth (multi-asset coverage) and depth (multi-methodology validation) for sophisticated investment decision-making in professional trading and investment management environments.

SAMPLE OUTPUT:

- hqm_commodities.csv
- hqm_bonds.csv