Feature Engineering Documentation

Crypto Investment Recommender System

📋 Table of Contents

🎯 Overview

What is Feature Engineering?

Feature engineering is the process of transforming raw data into meaningful metrics that help make better decisions. Think of it like turning raw ingredients into a recipe score.

Real-World Analogy:

- **Raw ingredients** → Market data (prices, volumes, rankings)
- **Cooking techniques** → Feature calculations (momentum, stability)
- **Recipe score** → Investment score (how good is this crypto?)

Purpose

Transform raw cryptocurrency market data into actionable investment recommendations by creating four key score categories:

1. **Momentum -** Is the price trending up?
2. **Value -** Is there growth potential?
3. **Stability -** How risky is it?

4. **Market Position** - How dominant is it?

📥 Input Data

Primary Data Source: `market_data_snapshot.csv`

Column	Description	Example	Why We Need It
`id`	Cryptocurrency identifier	"bitcoin"	Unique identifier
`current_price`	Current price in USD	$107,112	Base valuation
`market_cap`	Total value of all coins	$2.1 trillion	Shows market size
`market_cap_rank`	Ranking by market size	1	Shows dominance
`total_volume`	24h trading volume	$86 billion	Shows liquidity
`high_24h` / `low_24h`	Price range today	High: $108K, Low: $105K	Shows volatility
`price_change_percentage_24h`	Price change today	-2.05%	Recent momentum
`price_change_percentage_7d`	Price change this week	+5.3%	Short-term trend
`price_change_percentage_30d`	Price change this month	+15.2%	Medium-term trend
`ath` (All-Time High)	Highest price ever	$120,000	Recovery potential
`ath_change_percentage`	Distance from ATH	-15.09%	Growth opportunity
`circulating_supply`	Coins currently available	19.5M BTC	Supply economics
`total_supply`	Total coins that will exist	21M BTC	Scarcity measure

Secondary Data Source: `historical_crypto_data.csv`

Column	Description	Purpose
`timestamp`	Date/time of data point	Track price over time
`price`	Historical price	Calculate trends
`volume`	Historical trading volume	Assess liquidity patterns
`coin_id`	Cryptocurrency identifier	Link to market data

🔧 Feature Engineering Pipeline

Pipeline Flow

```
[1] Calculate Momentum Features (35% weight)
  ↓
[2] Calculate Value Features (25% weight)
  ↓
[3] Calculate Stability Features (20% weight)
  ↓
[4] Calculate Market Position Features (20% weight)
  ↓
[5] Combine into Investment Score (0-100)
  ↓
[6] Classify Risk Level (Conservative/Moderate/Aggressive)
  ↓
[7] Generate Recommendation (Strong Buy/Buy/Hold/Avoid)
  ↓
Final Output: Ranked Crypto List
```

📊 Detailed Feature Calculations

1. MOMENTUM FEATURES (35% of final score)

Goal: Measure price movement strength and direction

1.1 Momentum Score

Formula:

```
momentum_score = (24h_change × 0.30) + (7d_change × 0.40) + (30d_change × 0.30)
```

Why this weighting?

- **7-day change (40%) -** Most important; shows recent trend without daily noise
- **24-hour change (30%) -** Captures immediate momentum
- **30-day change (30%) -** Provides longer-term context

Example Calculation:

```
Bitcoin:
  24h change: -2.05%
  7d change: +5.3%
  30d change: +15.2%

momentum_score = (-2.05 × 0.30) + (5.3 × 0.40) + (15.2 × 0.30)
               = -0.615 + 2.12 + 4.56
               = 6.065%
```

Interpretation:

- **Positive score** → Upward momentum (good for buying)
- **Negative score** → Downward momentum (cautious)
- **Above 5%** → Strong momentum
- **Below -5%** → Weak momentum

1.2 Volume-to-Market-Cap Ratio

Formula:

```
volume_to_mcap_ratio = (total_volume / market_cap) × 100
```

Purpose: Measures liquidity - how easily can you buy/sell?

Example:

```
Bitcoin:
  Volume: $86,500,000,000
  Market Cap: $2,135,000,000,000

ratio = (86.5B / 2,135B) × 100 = 4.05%
```

Interpretation:

- **Above 10%** → High liquidity (easy to trade)
- **5-10%** → Good liquidity
- **Below 5%** → Lower liquidity (harder to trade large amounts)

1.3 Market Cap Momentum

Formula:

```
mcap_momentum = market_cap_change_percentage_24h
```

Purpose: Shows if the overall value is growing (better than just price)

Why it matters: A coin's price can rise, but if supply increases too, market cap momentum reveals true growth

---

2. VALUE FEATURES (25% of final score)

Goal: Identify undervalued assets with growth potential

2.1 Distance from All-Time High (ATH)

Formula:

```
distance_from_ath = |ath_change_percentage|
```

Purpose: How far below its peak price is this crypto?

Example:

```
Bitcoin:
  ATH: $120,000
  Current Price: $107,112
  ATH Change: -15.09%

distance_from_ath = 15.09%
```

Why it's useful:

- **0-10% from ATH** → Near peak, limited upside
- **10-35% from ATH** → Sweet spot! Recovery potential
- **35-60% from ATH** → Deep value or recovering
- **Above 60% from ATH** → Either troubled asset or extreme value

---

2.2 Recovery Potential Score

Formula (Logic-Based):

```python
def calculate_recovery_potential(distance_from_ath, momentum):
    if distance_from_ath < 5:
        return 40  # Too close to ATH, limited upside
    elif distance_from_ath <= 15:
        return 70 + (momentum × 0.5)  # Good position
    elif distance_from_ath <= 35:
        return 80 + (momentum × 0.3)  # SWEET SPOT
    elif distance_from_ath <= 60:
        return 60 + (momentum × 0.2)  # Recovering
    else:
        return 30 + (momentum × 0.1)  # Deep value or troubled
```

Real-World Logic:

- Assets 10-35% below ATH with positive momentum = best recovery potential
- Combines **value** (below ATH) with **momentum** (moving up)
- This is called "buying the dip" in trading

Example:

```
Bitcoin:
  Distance from ATH: 15.09%
  Momentum: +6.065%

recovery_potential = 70 + (6.065 × 0.5)
                   = 70 + 3.03
                   = 73.03 (out of 100)
```

2.3 Scarcity Score

Formula:

```
supply_ratio = circulating_supply / total_supply
scarcity_score = (1 - supply_ratio) × 100
```

Purpose: How much more supply can enter the market?

Example:

```
Bitcoin:
  Circulating: 19.5M
  Total: 21M

supply_ratio = 19.5 / 21 = 0.929
scarcity_score = (1 - 0.929) × 100 = 7.1%
```

Interpretation:

- **High scarcity (low %)** → Most supply already released (like Bitcoin)
- **Low scarcity (high %)** → More coins coming (potential inflation)

2.4 Price Range (24h Volatility Proxy)

Formula:

```
price_range_24h = ((high_24h - low_24h) / current_price) × 100
```

Purpose: How much did price swing today?

Example:

```
Bitcoin:
  High: $108,500
  Low: $105,900
  Current: $107,112

price_range = ((108,500 - 105,900) / 107,112) × 100
            = (2,600 / 107,112) × 100
            = 2.43%
```

Interpretation:

- **Below 3%** → Low daily volatility (stable)
- **3-7%** → Moderate volatility

- **Above 7%** → High volatility (risky)

---

3. STABILITY FEATURES (20% of final score)

Goal: Assess risk through volatility and consistency

3.1 Historical Volatility

Formula (from historical prices):

```
1. Calculate daily returns: return_t = (price_t - price_t-1) / price_t-1
2. Calculate standard deviation of returns
3. volatility_score = std_dev(returns) × 100
```

Step-by-Step Example:

```
Day 1: $100 → Day 2: $102 → Return = (102-100)/100 = 2%
Day 2: $102 → Day 3: $101 → Return = (101-102)/102 = -0.98%
Day 3: $101 → Day 4: $105 → Return = (105-101)/101 = 3.96%

Returns: [2%, -0.98%, 3.96%, ...]
Standard Deviation = 2.5%
volatility_score = 2.5
```

Interpretation:

- **Below 3%** → Low volatility (safe, boring)
- **3-7%** → Moderate volatility (balanced)
- **Above 7%** → High volatility (risky, exciting)

Why standard deviation? Standard deviation measures how spread out the returns are. Low = predictable, High = unpredictable.

3.2 Market Cap Stability Score

Formula:

```
max_mcap = highest_market_cap_in_dataset
mcap_stability_score = (log10(market_cap) / log10(max_mcap)) × 100
```

Why logarithm? Market caps range from millions to trillions. Log scale compresses this range for fair comparison.

Example:

```
Bitcoin Market Cap: $2,135,000,000,000 (2.135 trillion)
Max in dataset: $2,135,000,000,000

log10(2.135 trillion) = 12.33
log10(2.135 trillion) = 12.33

mcap_stability = (12.33 / 12.33) × 100 = 100 (most stable)
```

Interpretation:

- **80-100** → Large cap (very stable)
- **50-80** → Medium cap (moderate stability)
- **Below 50** → Small cap (less stable)

3.3 Combined Stability Score

Formula:

```
inverted_volatility = (max_vol - volatility_score) / max_vol × 100
stability_score = (inverted_volatility × 0.5) + (mcap_stability × 0.5)
```

Why invert volatility? Lower volatility = more stable, but we want higher scores to mean better. So we flip it.

Example:

```
Bitcoin:
  Volatility: 2.43% (low)
  Max volatility in dataset: 10%
  Market cap stability: 100

inverted_volatility = (10 - 2.43) / 10 × 100 = 75.7
stability_score = (75.7 × 0.5) + (100 × 0.5)
                = 37.85 + 50
                = 87.85
```

Interpretation:

- **Above 80** → Very stable (Conservative investment)
- **60-80** → Moderately stable
- **Below 60** → Less stable (Aggressive investment)

---

4. MARKET POSITION FEATURES (20% of final score)

Goal: Assess competitive position and dominance

4.1 Rank Score

Formula:

```
max_rank = highest_rank_in_dataset (e.g., 20)
rank_score = ((max_rank - market_cap_rank + 1) / max_rank) × 100
```

Why this formula? Rank 1 should get the highest score, rank 20 the lowest. We invert the ranking.

Example:

```
Bitcoin:
  Rank: 1
  Max rank: 20

rank_score = ((20 - 1 + 1) / 20) × 100
           = (20 / 20) × 100
           = 100

Tron:
  Rank: 9
  Max rank: 20

rank_score = ((20 - 9 + 1) / 20) × 100
           = (12 / 20) × 100
           = 60
```

4.2 Market Dominance

Formula:

```
total_market_cap = sum of all cryptos in dataset
market_dominance = (asset_market_cap / total_market_cap) × 100
```

Purpose: What percentage of the total market does this crypto represent?

Example:

```
Bitcoin: $2.135 trillion
Total market (20 cryptos): $3.314 trillion

market_dominance = (2.135 / 3.314) × 100 = 64.4%
```

Interpretation:

- **Above 50%** → Market leader (Bitcoin)
- **10-50%** → Major player
- **Below 10%** → Smaller player

4.3 Liquidity Score

Formula:

```
max_volume_ratio = 95th percentile of volume_to_mcap_ratio
liquidity_score = (volume_to_mcap_ratio / max_volume_ratio) × 100
```

Why 95th percentile? Avoids outliers. Some cryptos have abnormally high volume ratios that would skew the scale.

Example:

```
Bitcoin volume ratio: 4.05%
95th percentile: 15%

liquidity_score = (4.05 / 15) × 100 = 27
```

Interpretation:

- **Above 70** → Extremely liquid

- **30-70** → Good liquidity
- **Below 30** → Lower liquidity

4.4 Market Tier Classification

Logic:

```
if market_cap_rank <= 5:
    tier = "Blue Chip"
elif market_cap_rank <= 20:
    tier = "Large Cap"
elif market_cap_rank <= 50:
    tier = "Mid Cap"
else:
    tier = "Small Cap"
```

Purpose: Quick categorization for portfolio allocation

Real-World Equivalent:

- **Blue Chip** = Like Apple, Microsoft (safest)
- **Large Cap** = Like Netflix, Adobe
- **Mid Cap** = Like Etsy, Zoom
- **Small Cap** = Like emerging startups

🎯 Investment Score Formula

Final Composite Score

Formula:

```
investment_score =
    (momentum_score_normalized × 0.35) +
    (recovery_potential_normalized × 0.25) +
    (stability_score × 0.20) +
    (rank_score_normalized × 0.20)
```

Why These Weights?

Component	Weight	Reasoning
Momentum	35%	Most important - "the trend is your friend" in trading
Value/Recovery	25%	Growth potential - buying undervalued assets
Stability	20%	Risk management - avoiding losses
Market Position	20%	Safety - established players are less risky

Total = 100%

Normalization Process

Why normalize? Different features have different scales (momentum might be -10 to +30, rank_score is 0-100). Normalization puts everything on the same 0-100 scale for fair comparison.

Formula:

```
normalized_value = ((value - min_value) / (max_value - min_value)) × 100
```

Example:

```
Momentum scores in dataset: [-5, 0, 3, 6, 10, 15]
Min = -5, Max = 15

Bitcoin momentum = 6
normalized = ((6 - (-5)) / (15 - (-5))) × 100
           = (11 / 20) × 100
           = 55
```

Final Score Interpretation

Score Range	Meaning	Action
75-100	Excellent opportunity	Strong Buy
60-74	Good opportunity	Buy
45-59	Decent option	Moderate Buy
30-44	Neutral	Hold
0-29	Poor outlook	Avoid

⚠️ Risk Classification

Classification Logic

```python
def determine_risk_level(volatility, market_cap_rank, market_cap):
    if rank <= 10 and volatility < 5:
        return "Conservative"
    elif rank <= 20 and volatility < 10:
        return "Moderate"
    else:
        return "Aggressive"
```

Risk Categories

Conservative (Low Risk)

- **Criteria:** Top 10 rank + Low volatility (< 5%)
- **Examples:** Bitcoin, Ethereum, Tether
- **Suitable for:** Risk-averse investors, retirement funds
- **Expected returns:** Lower but stable

Moderate (Medium Risk)

- **Criteria:** Top 20 rank + Moderate volatility (< 10%)
- **Examples:** Cardano, Solana, Polkadot
- **Suitable for:** Balanced portfolios
- **Expected returns:** Moderate with some fluctuation

Aggressive (High Risk)

- **Criteria:** Lower rank OR High volatility (> 10%)
- **Examples:** Small-cap tokens, new projects
- **Suitable for:** Risk-tolerant investors, speculation
- **Expected returns:** High potential but high risk

📤 Output Data

Output File: `crypto_features.csv`

Columns Added by Feature Engineering:

Column	Type	Range	Description
`momentum_score`	Float	-30 to +30	Weighted price momentum
`recovery_potential`	Float	0-100	Growth opportunity score
`stability_score`	Float	0-100	Risk/volatility assessment
`rank_score`	Float	0-100	Market position score
`investment_score`	Float	0-100	**FINAL SCORE** for ranking
`risk_level`	String	Categorical	Conservative/Moderate/Aggressive
`recommendation`	String	Categorical	Strong Buy/Buy/Moderate Buy/Hold/Avoid
`reasoning`	String	Text	Human-readable explanation
`volatility_score`	Float	0-20	Price fluctuation measure
`volume_to_mcap_ratio`	Float	0-100	Liquidity indicator
`market_tier`	String	Categorical	Blue Chip/Large Cap/Mid Cap/Small Cap
`distance_from_ath`	Float	0-100	% below all-time high
`liquidity_score`	Float	0-100	Trading ease measure
`market_dominance`	Float	0-100	% of total market

Sample Output

```
Rank 1: Bitcoin
  Investment Score: 79.2
  Risk Level: Conservative
  Recommendation: Strong Buy
  Reasoning: Market leader, low risk, positive momentum (+6.1%).

Rank 2: Ethereum
  Investment Score: 76.8
  Risk Level: Conservative
  Recommendation: Strong Buy
  Reasoning: Established player, low risk, high liquidity.
```

🔍 Quality Checks

Validation Performed

1. **Missing Values:** All critical fields filled or default values assigned
2. **Outliers:** Handled using percentile-based normalization (95th percentile)
3. **Data Types:** All numerical fields validated
4. **Score Ranges:** All scores clipped to 0-100 range
5. **Consistency:** Ranks match investment scores (higher score = higher rank)

Error Handling

```python
# Example checks in the code:
- .fillna(0) for missing momentum data
- .replace(0, np.nan) to avoid division by zero
- .clip(0, 100) to ensure scores stay in valid range
- try-except blocks for API failures
```

📚 References & Methodology

Statistical Methods Used

1. **Weighted Average:** Combining multiple time periods for momentum
2. **Standard Deviation:** Measuring volatility
3. **Logarithmic Scaling:** Handling large number ranges (market caps)
4. **Min-Max Normalization:** Standardizing features to 0-100 scale
5. **Percentile-Based Outlier Handling:** Using 95th percentile thresholds

Financial Concepts Applied

1. **Factor Investing:** Multi-factor model (momentum, value, quality)
2. **Mean Reversion:** Recovery potential from ATH
3. **Risk-Return Tradeoff:** Balancing volatility with returns
4. **Market Efficiency:** Using market cap as safety proxy
5. **Liquidity Premium:** Valuing tradeable assets higher

Real-World Parallels

This model is inspired by:

- **Morningstar Star Ratings** (mutual fund ratings)
- **Fama-French Factor Models** (academic finance)
- **Quantitative Trading Strategies** (algorithmic trading)
- **Credit Rating Systems** (S&P, Moody's methodology)

✅ Summary

What We Built

A systematic, transparent, and defensible cryptocurrency investment recommendation system that:

1. Transforms 21 raw market metrics into 15 engineered features
2. Combines them into a single 0-100 investment score
3. Classifies risk levels based on volatility and market cap
4. Generates actionable buy/hold/avoid recommendations
5. Provides human-readable reasoning for each recommendation

Why It Works

- **Explainable:** Every score can be traced back to specific calculations
- **Balanced:** No single metric dominates the decision
- **Practical:** Based on proven financial theories
- **Scalable:** Works for any cryptocurrency with market data
- **Reproducible:** Same inputs always produce same outputs

Limitations Acknowledged

- Historical performance doesn't guarantee future results
- Market conditions change rapidly in crypto
- Model assumes rational market behavior
- Does not account for external events (regulations, hacks, news)

- Best used as one input among many for investment decisions

Document Version: 1.0

Last Updated: October 23, 2025

Author: Christianah - AI/ML Engineer Applicant