

DCF Valuation System

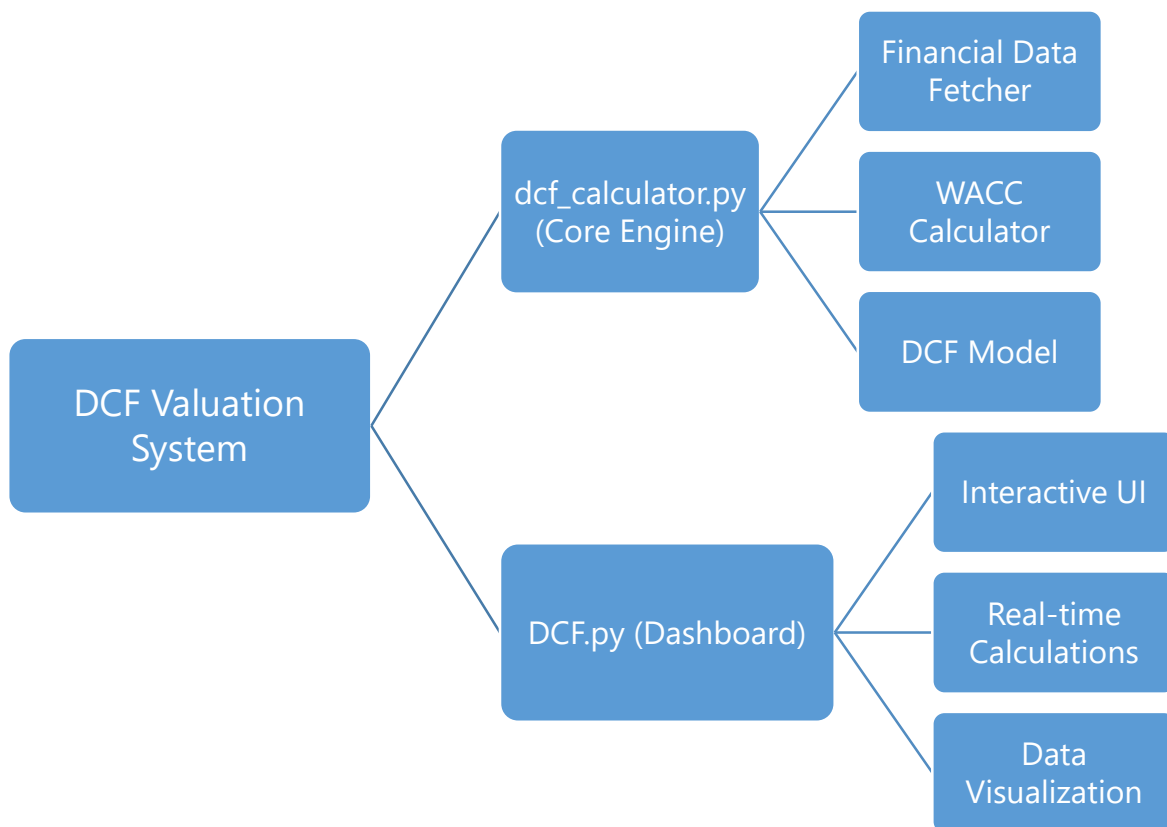
Documentation

Discounted Cash Flow valuation system provides comprehensive financial analysis and stock valuation capabilities.

The system consists of two main components:

1. **dcf_calculator.py** - Core calculation engine for DCF analysis
2. **DCF.py** - Interactive Dash web application for visualization and analysis

System Architecture



Core Components (dcf_calculator.py)

1. Financial Data Fetcher Class

Purpose: Retrieves and processes financial data from Yahoo Finance API.

Key Methods:

- a. `get_financial_statements()`: Fetches income statement, balance sheet, and cash flow data
- b. `get_key_metrics()`: Extracts essential financial metrics from statements
- c. `get_market_data()`: Retrieves market-related data (market cap, beta, current price)

Data Retrieved:

- a. Income Statement: Revenue, EBIT, EBITDA, Net Income, Tax Expense
- b. Balance Sheet: Total Debt, Cash, Working Capital components
- c. Cash Flow: Operating Cash Flow, Capital Expenditure, Depreciation
- d. Market Data: Market Cap, Shares Outstanding, Beta, Current Stock Price

2. WACC Calculator Class

Purpose: Calculates the Weighted Average Cost of Capital using the CAPM model.

Formula: $WACC = (E/V \times Re) + (D/V \times Rd \times (1-T))$

Where:

E = Market value of equity

D = Market value of debt

V = Total value (E + D)

Re = Cost of equity

Rd = Cost of debt

T = Tax rate

Cost of Equity Calculation: $Re = Rf + \beta \times (Rm - Rf)$

Rf = Risk-free rate (default: 4.5%)

β = Stock beta

Rm - Rf = Market risk premium (default: 6.5%)

3. DCF Model Class

Purpose: Orchestrates the complete DCF valuation process.

Key Methods:

- a. `calculate_historical_averages()`: Computes 3-year historical averages for key metrics
- b. `project_cash_flows()`: Projects future free cash flows for specified years
- c. `calculate_terminal_value()`: Estimates terminal value using perpetual growth model
- d. `calculate_dcf_valuation()`: Performs complete DCF analysis

Valuation Process:

- a. Historical Analysis: Calculate 3-year averages for growth rates, margins, and ratios
- b. Cash Flow Projection: Project future free cash flows based on historical trends
- c. Terminal Value: Calculate terminal value using perpetual growth model
- d. Present Value: Discount all cash flows to present value using WACC
- e. Equity Value: Subtract net debt from enterprise value
- f. Per Share Value: Divide equity value by shares outstanding

Dashboard Application (DCF.py)

User Interface Components

1. Input Panel

- a. Stock Ticker: Enter any publicly traded stock symbol
- b. Risk-Free Rate: Adjustable (default: 4.5%)
- c. Market Risk Premium: Adjustable (default: 6.5%)
- d. Terminal Growth Rate: Adjustable (default: 2.5%)
- e. Projection Years: 3-10 years (default: 5)

2. Key Metrics Display

- a. Current Price: Real-time stock price
- b. DCF Fair Value: Calculated intrinsic value per share
- c. Upside/Downside: Percentage difference from current price
- d. WACC: Weighted Average Cost of Capital
- e. Enterprise Value: Total company value

3. Interactive Visualizations

a. Revenue and FCF Projections:

1. Bar chart showing projected revenue growth
2. Line chart overlaying free cash flow projections
3. Dual y-axis for different scales

b. Valuation Breakdown:

1. Waterfall chart showing enterprise value components
2. Present value of cash flows vs. terminal value
3. Visual breakdown of value sources

c. WACC Components:

1. Pie chart showing capital structure (debt vs. equity)
2. Bar chart displaying cost components
3. Visual representation of financing mix

d. Sensitivity Analysis:

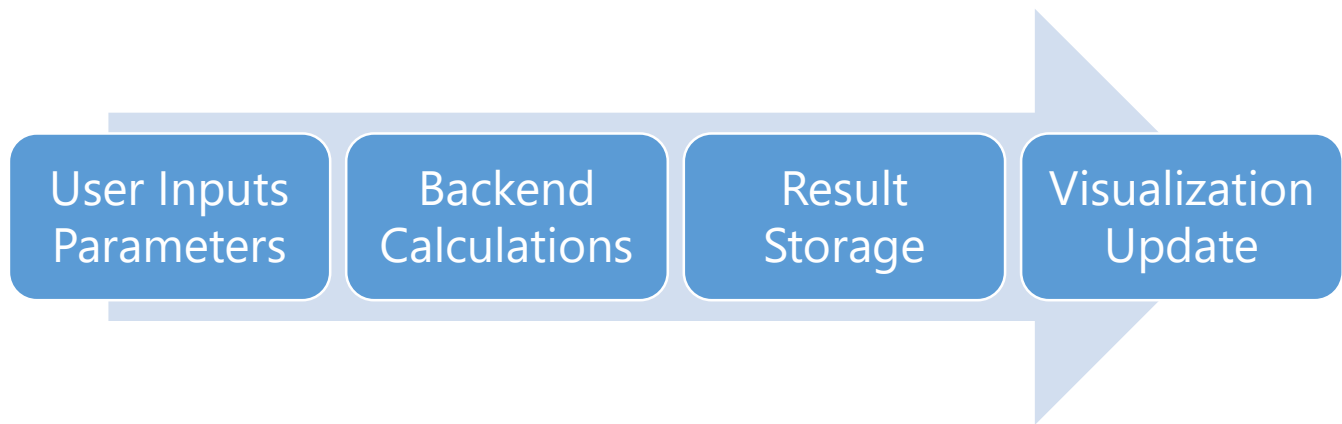
1. Heatmap showing value sensitivity to WACC and terminal growth changes
2. Color-coded impact visualization
3. Risk assessment tool

Technical Implementation

Framework and Libraries:

1. Dash: Web application framework
2. Plotly: Interactive charting and visualization
3. yfinance: Financial data retrieval
4. pandas/numpy: Data manipulation and analysis

Data Flow



Usage Instructions

Install required packages

`pip install dash plotly pandas numpy yfinance`

Run the dashboard

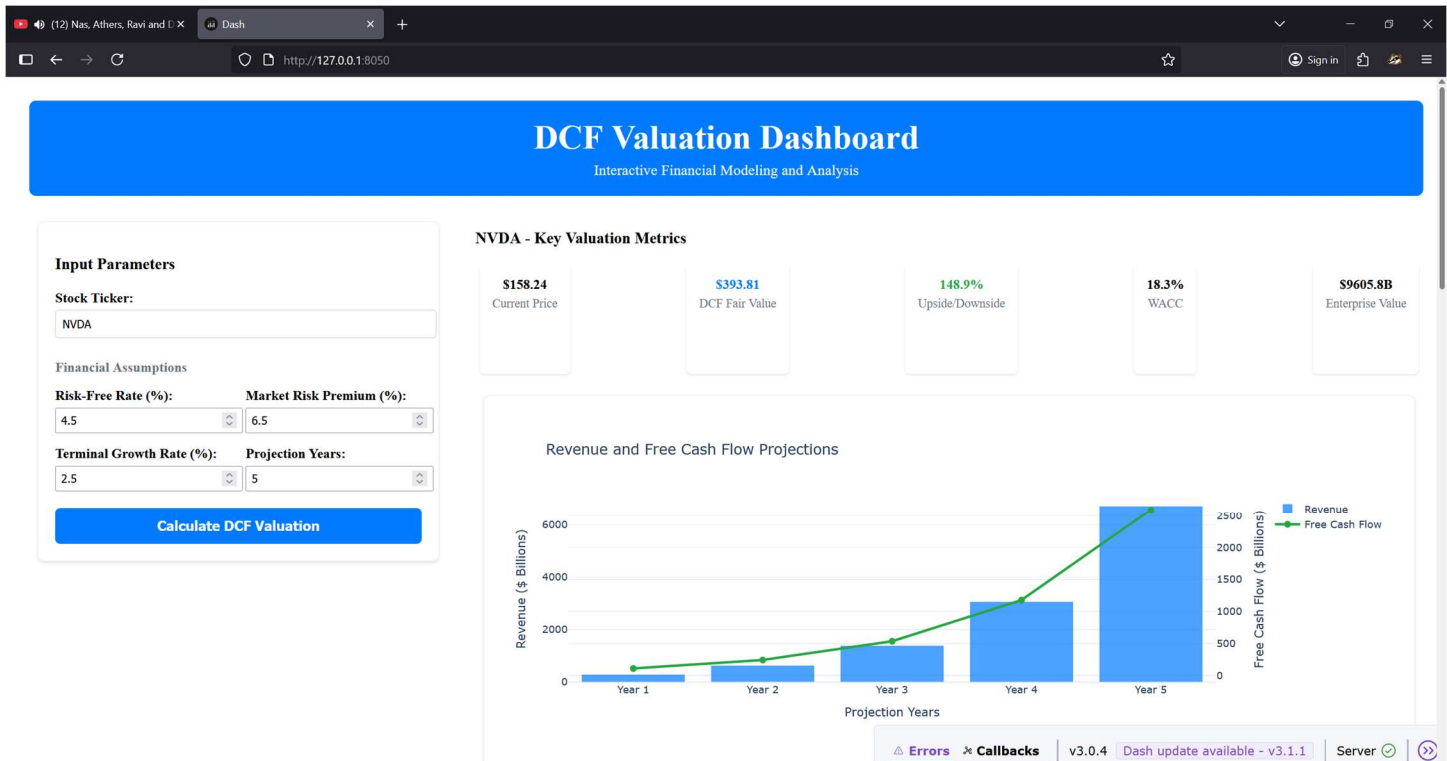
`python "DCF.py FILE PATH"`

```
Administrator: Anaconda Prompt - python "C:\Users\manth\OneDrive\Desktop\DCF.py"
(base) C:\Windows\System32>python "C:\Users\manth\OneDrive\Desktop\DCF.py"
Dash is running on http://127.0.0.1:8050/

* Serving Flask app 'DCF'
* Debug mode: on
```

Adjustable Parameters

- **Risk-Free Rate:** Update based on current treasury yields
- **Market Risk Premium:** Adjust for market conditions
- **Terminal Growth Rate:** Industry-specific growth assumptions
- **Projection Period:** Extend or shorten forecast horizon



Best Practices

1. Input Selection

- Use realistic, conservative assumptions
- Cross-reference with industry benchmarks
- Consider economic cycle impacts
- Validate with multiple valuation methods

2. Interpretation

- Focus on ranges rather than precise values
- Consider margin of safety (e.g., 20-30% discount)
- Perform sensitivity analysis

- Compare with peer valuations

3. Model Maintenance

- Regular assumption updates
- Historical accuracy tracking
- Model validation against market prices
- Continuous refinement based on performance

Troubleshooting

- 1. Data Not Loading:** Check internet connection and ticker symbol
- 2. Calculation Errors:** Verify input parameters are within valid ranges
- 3. Visualization Problems:** Ensure all required libraries are installed
- 4. Performance Issues:** Consider reducing projection years for complex calculations

This documentation provides a comprehensive guide to understanding and using the DCF valuation system. For additional support or advanced customization, refer to the inline code comments and financial modeling best practices or contact Manthan Gehlot CMT, CFTe at LinkedIn or email at manthangehlot66@gmail.com