

Real Estate Portfolio Optimization

sdmay19-07

<http://sdmay19-07.sd.ece.iastate.edu/>

Advisor Chinmay Hendge

Client Principal Financial Group

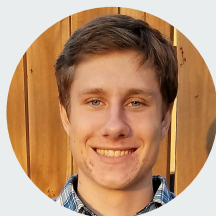
Meet the Team



**Blake
Roberts**

Project Lead /
Backend

Software
Engineering



**Kevin
Johnson**

Quality Control /
Frontend

Computer
Engineering



**Nickolas
Moeller**

Report Manager /
Backend

Software
Engineering



**Leelabari
Fulbel**

Meeting Facilitator /
Frontend

Software Engineering



Colton Goode

Meeting Scribe /
Backend

Computer Engineering,
Management of
Information Systems

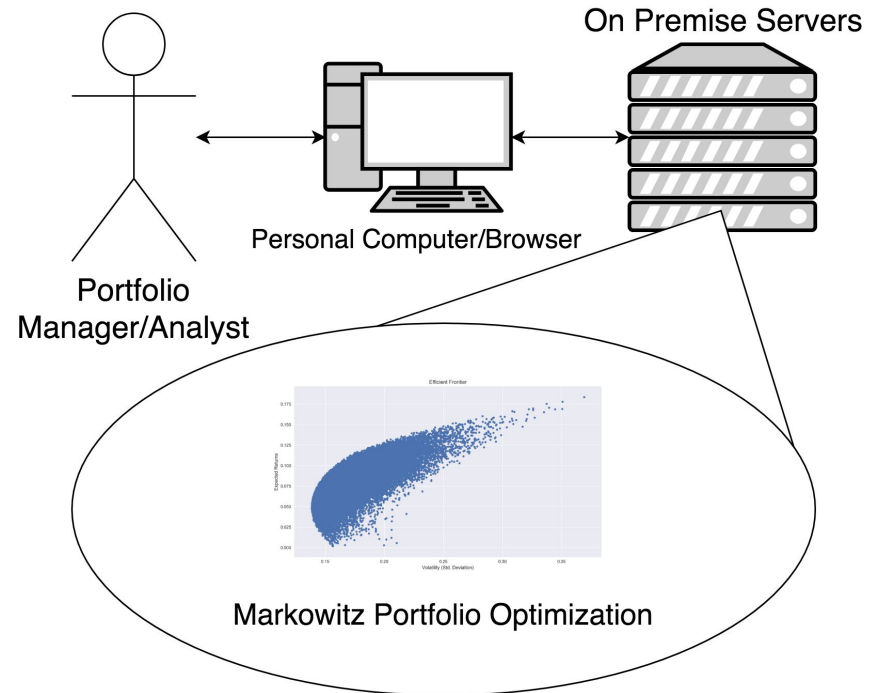


Problem Statement

- Portfolio managers (PM's) must decide the composition of assets within their real estate portfolio
- Asset, market and portfolio level analysis helps form this decision
- Principal utilizes a third party, Costar, to gain portfolio insight
 - Reports are long, inefficient, non-customizable, and pricy

Conceptual Sketch

- Input market data and portfolio holdings
- Perform Markowitz portfolio optimization
- Output analytical data visualization



Requirements



Functional Requirements

User can ...

1. Generate optimal portfolios via configurable constraints such as market and property type
2. Visualize efficient frontiers compared to markets, property types, and the current portfolio
3. Visualize current holdings by geography, property type, expected return, and risk
4. Visualize differences between optimal and current holdings in maps, plots, and other charts
5. Receive recommend buy and sell decisions given the current portfolio holdings
6. Share results via file export and/or email



Non-functional Requirements

- The system will use only open source libraries and frameworks
- Principal data must not pass through non-vetted third party systems
- Optimization must take no longer than 5 seconds to calculate



Constraints and Considerations

- Principal has servers to host internal applications
- Future project will be maintained by Principal's data science team
- The application must be easy to use and should not require training

System Design



Detailed Design

Input specification:

- NCREIF market data
- Portfolio holdings

Output specification:

- Optimized portfolio
- Efficient frontier

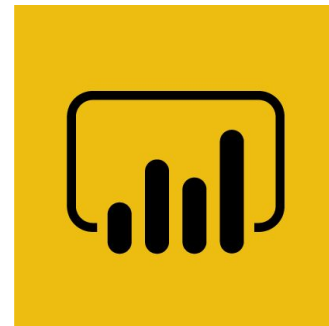
UI and software specifications:

- JavaScript (runtime compiled)
- Modern browsers (Chrome)



Software/Technology Platforms Used

- Python
- Flask
- SQLite
- Dash
- Plotly
- Power BI

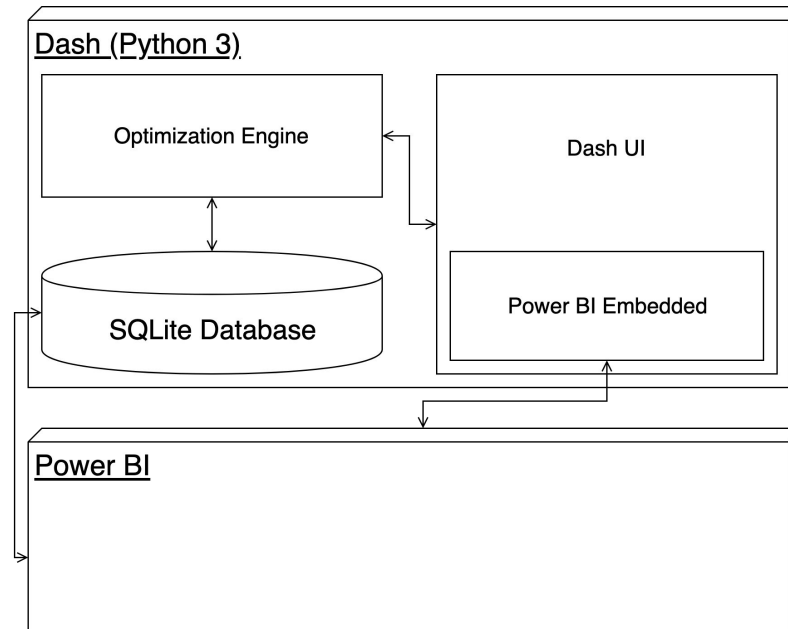




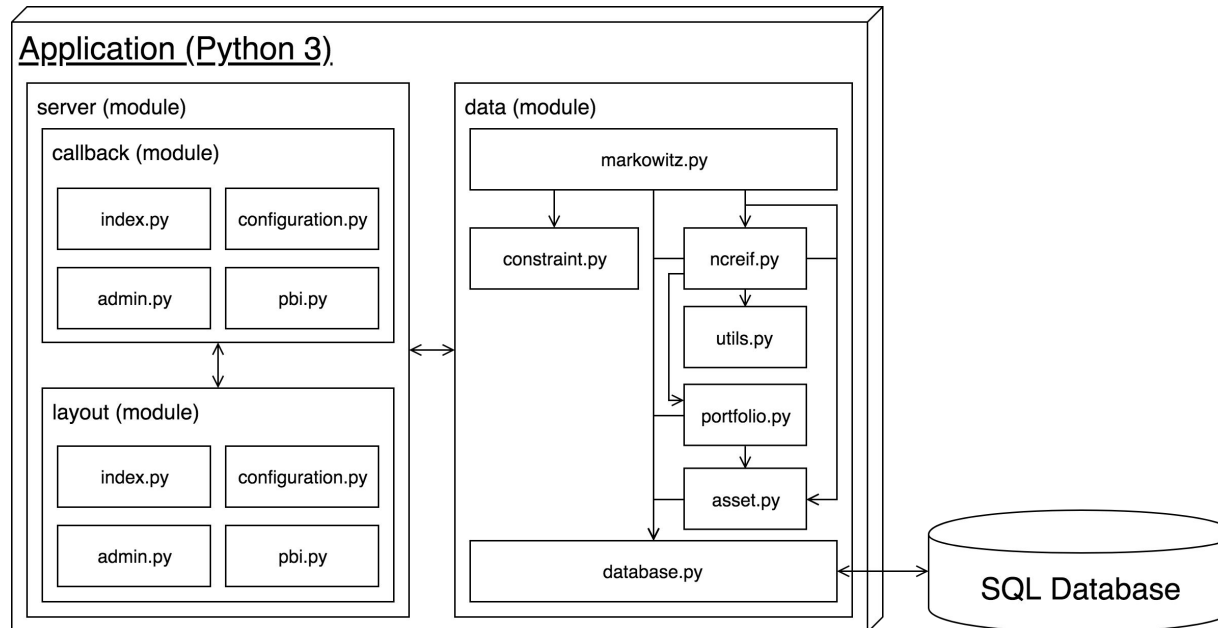
System Description / Operating Environment

- Operating System Agnostic
- Web application written in Python utilizing the Flask framework
- Client views built with Dash and Power BI
- SQLite or on premise SQL Database

Architecture Block Diagram



Software Block Diagram





User Interface

Dash, a Python library built on top of Plotly, compiles JavaScript data visualizations (i.e. charts and graphs)

The application consists of four major components:

- uploading portfolios
- creating constraints
- viewing the current and optimized portfolio
- comparisons and Efficient Frontier

Home	Choose Portfolio and Configure Constraints	Optimization Results	Application Administration
------	--	----------------------	----------------------------

Summary

This application is meant to provide market level analysis for portfolio advisors. By uploading a portfolio and inputting constraints, the application will perform Markowitz portfolio optimization. The optimization data will be sent to Power BI for data visualization.

How to Use the Application:

- There are four parts (or tabs) which make up the app.
- The first part is this current tab, the landing page.
- The second is the "Configure Portfolio and Constraints" part. Here you will be able to upload a portfolio and customize various optimization constraints.
- After hitting "Run Optimization", the "Optimization Results" tab will embed a Power BI report which displays the results of your optimization.
- The final tab, "Application Administration", is extra functionality mostly used for uploading new quarterly market data. This application uses NCREIF data to calculate covariance matrices and estimate expected return.

Landing Page with Instructions

Select an Existing Portfolio

Upload a New Portfolio

Drag and Drop or [Select Files](#)

Configure Constraints

Market Constraints



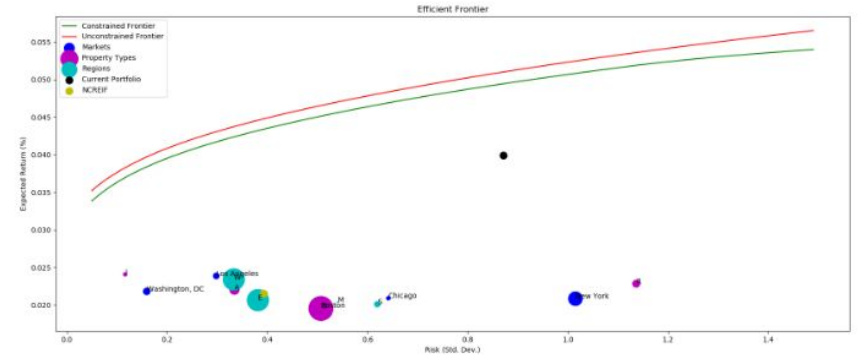
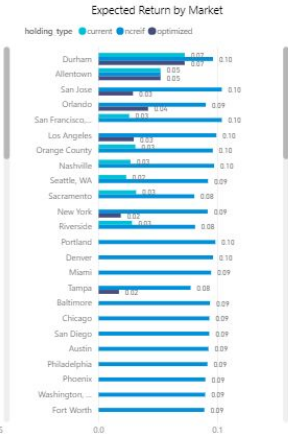
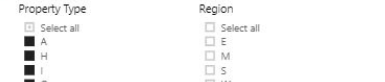
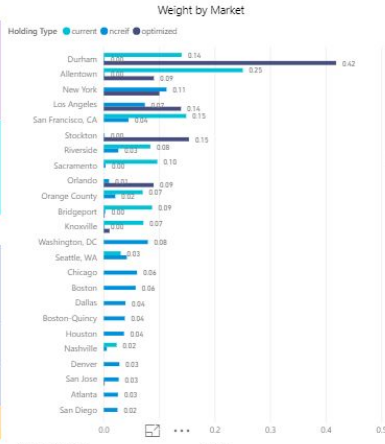
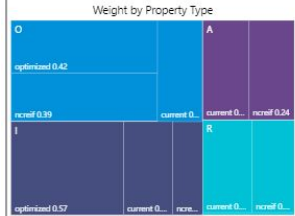
Property Type Constraints



Market Property Type Constraints

Region Constraints

Portfolio Upload & Optimization Constraints



Power BI Embedded Visual

Home	Choose Portfolio and Configure Constraints	Optimization Results	Application Administration
------	--	----------------------	----------------------------

Admin Page

NCREIF File Upload (.csv or .xlsx)

Drag and Drop or [Select Files](#)

Edit Portfolio

Select a portfolio ▼

Admin Page

Testing and Evaluation



Test Plan

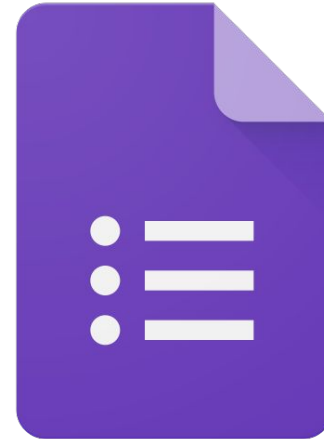
- Utilize Python's unittest package
- User testing in Spring 2019





User Testing

- Coordinated tests with Principal
- Utilized Google Forms
- Responses immediately implemented





Evaluation & Future

- Overall success
- Learned a lot
- Experience
- Already used by Principal
- Expected Returns

Questions?