Technology Review

Asset Portfolio Performance

Team Members:
Will Bishop
Vivek Pagadala
Vamsy Atluri

Background

1. Allow users to select their choice of asset allocation models (stock indices/bonds split) and a risk measure.
2. Give the risk vs reward value for each selected portfolio.
3. Allow users to see this performance over a time period.

Visualization

Dash vs. Bokeh

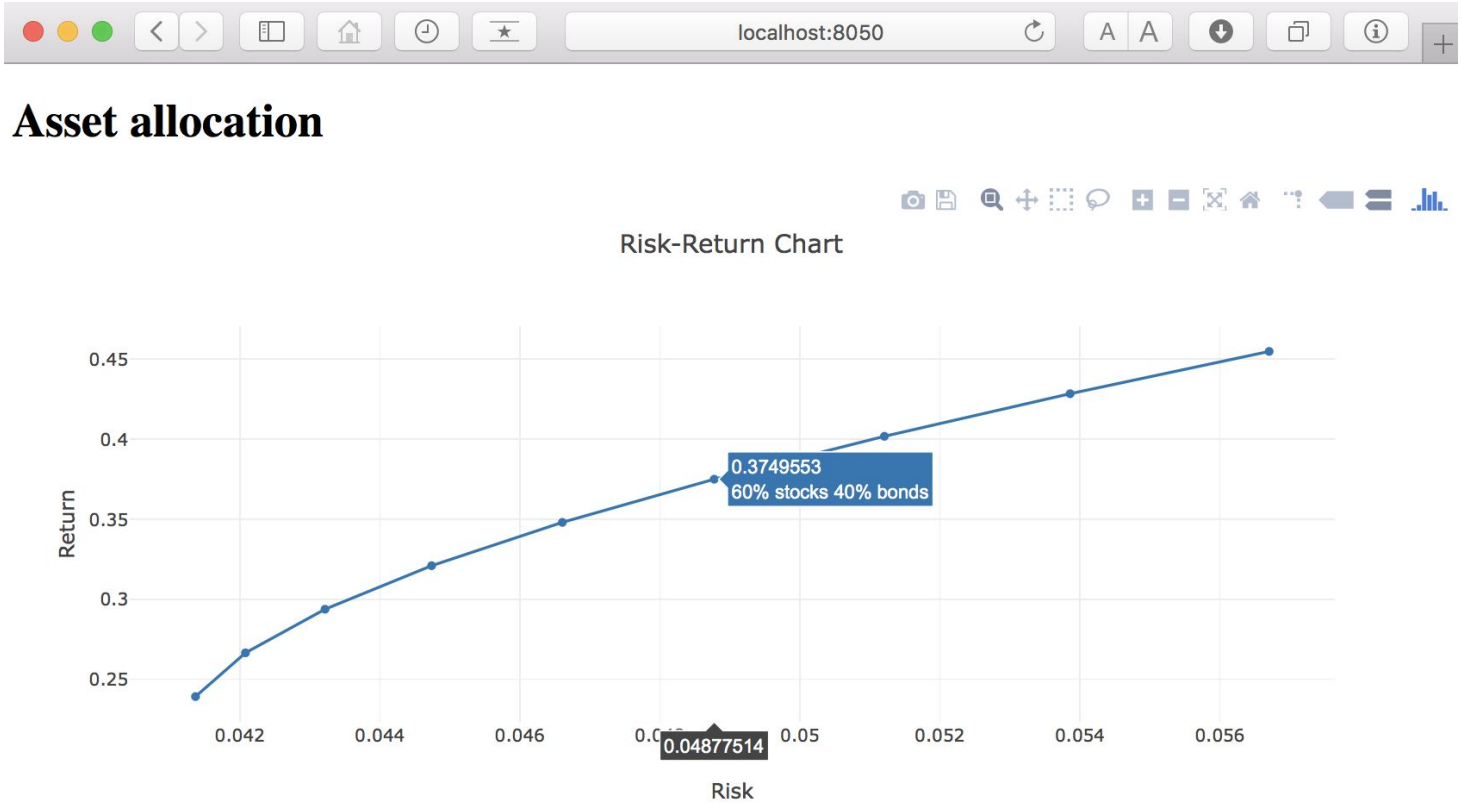
Need for Visualization?

- Project compares multiple asset allocation portfolios on different measures of risk and reward
- Goal: show user a graph of the performance of different portfolios
- Requires a library that can capture user input and show plots and visualizations
- We want to enable the user to interact as much as possible with the visualization and play around with different parameters

Why Dash?

- Dash and Bokeh are the two dashboard frameworks we considered
- Advantages of Dash over Bokeh:
  - Bokeh needs some JavaScript for graph interactions, has a bit of a learning curve.
  - Dash, when used with plotly, requires only Python
  - Dash also makes it easy to add components since it uses React on the frontend and provides python wrappers to HTML components
  - This ties in with our goal of making the output visualization as interactive as possible

Initial draft of the app using Dash



Data Storage

Pandas, SQLite vs. MySQL/SQL Server

Need for Data Storage?

- We have one .csv file for each stock and bond index we are tracking
- Depending on user selections like risk measure and portfolio split, our function calculates the corresponding risk vs reward values and plots them on a dashboard
- CSV files are convenient but not meant for long term data storage, especially when building an application that can be expanded on in the future with additional data.
- Our goals merit the use of a relational database that is easy to work with and integrates well with python.

Pandas

- Pandas is a library that allows in-memory manipulation of tabular data
- Compared to usual SQL databases, pandas has an extensive set of functions that make manipulating data much easier.
- Pandas also has native support for time-series data and provides functions such as resampling.
- It works well with various data formats: CSV files, SQLite database files, etc.
- In a nutshell, pandas gives all the basic data manipulation functions while making operations faster and convenient. This makes working with data much easier.

SQLite

- SQLite is an embedded database we can deploy along with our application
- Compared to SQL server or MySQL, SQLite is fast and doesn’t need separate installation
- Size of our data does not warrant a full fledged RDMS
- No user writes in our application
- SQLite supports multiple reads at the same time making it ideal for our use case
- Ability to ship the code as an application in itself

Cloud Deployment

AWS vs. Azure

Why Cloud Deployment?

- At present, we are running the program on our own workstations and using Dash presents us the output visualization at a link on our local system.
- While this is still ideal in the sense users can run our script on their devices and view the correct output, we feel enabling users with lesser technical know-how to use the application freely is more desirable.

Why AWS?

- AWS (Amazon Web Services) provides the infrastructure and tools to enable users deploy their applications on the cloud.
- This enables developers to host their applications on AWS' servers and scale it such that thousands of end users can access it at the same time.
- Compared to Azure, AWS is the one we have most experience with and feel is easier to work on. Documentation is much cleaner and extensive.
- There are also no Microsoft Services we are using in our project (like SQL Server etc.) which cancel out any advantages that Azure may offer.
- Ability to run AWS EC2 instances for free as part of its free tier offerings (for one year after signing up at least).