Quant Research Rapid Prototyping Library

Design Document

sdMay19-06

Client: Principal Financial

Advisor: Srikanta Tirthapura

Doh Yun Kim
Drake Mossman
Gabriel Klein
Josiah Anderson
Nathan Schaffer
Jacob Richards
Team Email: sdmay19-06@iastate.edu

Table of Contents

1 Introduction

1.1 Acknowledgement

Team 06's Faculty advisor: Srikanta Tirthapura
Team 06's client: Principal Financial, primarily Benjamin Harlander and Vishnu Vemuru
Interviewed Data Scientists: Josh Zimmerman, Krisoye Smith, Ryan Lam, Markus Sauter, Q Mabasa, Yaoliang He.
We would like to thank Srikanta Tirthapura for being our advisor and helping us out.

1.2 Problem and Project Statement

Principal Financial recently created a team of data scientists to work within their Global Investments Department to analyze equity data and help the company make better financial decisions. This team does not have a unified or consistent way of aggregating data and running models on data. This causes losses in time and efficiency.

Our task is to research effective tools and procedures for the data science team, and create a unified application that can be used by all members of the team to run prediction models on stock equity data.

1.3 Operational Environment

The intended operating environment for this application will be the personal work computers used by the data scientists at Principal. The software environment for this application is something that is still flexible and needs to be established.

1.4 Intended Users and Uses

The intended users of this project are the 11 data scientists already employed at Principal Financial and any future data scientists that will work alongside them. This means that this tool must be easy to grasp and must be intuitive for those experienced in data science. The uses of this product should be similar for all users: to take stock equity data and use it to model the most likely trajectory of those stocks.

1.5 Assumptions and Limitations

Assumptions:

1. All data will come from Factset
2. Users will have moderate experience with either Python or R

Limitations:

1. Most data scientists only know one of Python or R
2. Takes in data in one set format
3. Outputs data in a specific format

1.6 Expected End Product and Deliverables

- R Library Focusing on preprocessing of data

  Our primary deliverable will be a library to be used by anyone on Principal's data science team, but specifically geared towards interns and students they work with. Because one of

our primary objectives is to save our users time, the library will be built for speed of processing and benchmarked against alternatives. It will likely be divided into multiple groups of functionality, the details of which will be specified further in future iterations of our project. Because of the evolving nature of the data science domain, we will need to design our library to be extensible so that the more experienced members of Principals team can update it as warranted. Much of the internals will be built upon existing libraries and already written code, modified as reused as we see fit.

- Package / Method Roxygen Documentation

  An essential deliverable for our project will be documentation our the functions we create. Our target users aren't very proficient with programming, and therefore will not be able to understand and use our libraries unless they are written at a high level and are ultimately intuitive. Along with standard documentation of performance of each method and parameter, it will likely be beneficial to give some examples of basic use cases for the utmost clarity.

- Process Documentation

  An overarching description of how our library will fit into Principal's current work processes will give our clients the uniformity they are looking for within their department. This description will be written in a non-technical manner and be designed with a managerial intent in mind. It should describe how the usage of our library fulfills many of the currently manual procedures for ensuring correctness of predictive models. Through our iterations of this deliverable, we will interface with our point persons and higher-up employees within Principal to make sure its effect is as planned.

- Modification Directive

  In order for our planning and design to have maximal impact on the future of our library, we will document directives for future software designers modifying the code we write. This will contain technical specifications of any architectural considerations and more detailed descriptions for intent of our libraries.

2. Specifications and Analysis

2.1 Proposed Design

So far our group has mainly been researching our client's situation to establish their needs by interviewing employees. We've carried out several interviews with various kinds of employees and found that their workflow is in serious need of automation for the less skilled employees, as well as for the many students and interns they work with. Specifically it would be helpful if we could automate any of the following tasks:

- Retrieving and formatting data
- Preprocessing data for the model
- Constructing accurate predictive models
- Postprocessing the model for validation
- Visualizing the results

As constructing and validating models will likely take significant statistical knowledge, we plan on focusing on one of the other areas.

What follows below are a few approaches we've discussed for automating this work. As of now the approach we choose and the task we choose to automate will influence the functional requirements and non-functional requirements of the project. We are close to reaching a consensus with our client on which of these to choose, after which we will be able to create these.

Python Library

The library will consist of high-level functions and objects with a variety of parameters able to perform the necessary data manipulations. This enables the user to create and test models using very few lines of code.

Strengths:
Python is a common general purpose language
Python is already used by some of the client's employees
Flexible functionality using object members and function parameters
Easily extendable by adding new calls
Easy to install and distribute

Weaknesses:
High level of parameterization needed could require verbose function calls or object initialization
Requires users to learn a new set of functions to interface with

Taking this approach, users will have to learn the set of functions we provide, which is true for really any solution we come up with for automating processes. Similarly, there may be significant explicit configuration required resulting in verbose code, but this is again true for any approach that wants to provide that level of customization.

On the other hand, Python is a powerful general purpose language with significant popularity in data science. It's faster than R, meaning the overall runtime of scripts using the library will be reduced. Additionally the library will be easily integratable into the client's current workflow, as it simply replaces large blocks of code with a single line that calls the library to do the same manipulations.

The library will provide an object type that can be loaded with data through a method call. Once initialized as such, other functions can then be called with the data object to perform manipulations and transformations. Due to the enormous amount of customization required for the various data manipulations, some generalized functions would require tens or hundreds of arguments to specify. To cut down on this, we'll offer various more specific calls for different classes of manipulations and keep the number of arguments on a small scale.

The internals of the library will make heavy use of the other libraries our client is already making use of including numpy, pandas, sklearn, and matplotlib. Each of these already implements a lot of very useful functionality, so our library will mainly be focused on combining them to perform larger tasks.

R Library

We considered meeting our objective by writing a library of data manipulation functionality implemented in R. The library would consist of high-level functions and objects with a variety of parameters able to perform the necessary processes using single lines of code.

Strengths:
R is a very common language used by data scientists
R is already used by some of the client's employees
Flexible functionality using object members and function parameters
Easily extendable by adding new calls
Easy to install and distribute

Weaknesses:
High level of parameterization needed could require verbose function calls or object initialization
R is relatively slow at performing the required processes
Requires users to learn a new set of functions to interface with

This solution seems weaker than the Python library mostly due to R's lackluster performance in data manipulations. Many of the processes required take a significant amount of time to complete depending on the amount of data to process and the type of model being tested. The unfortunate lack of speed when using R would result in much longer runtime, which is counter to the project's goal of saving time through automation. Additionally we found that using Python would be a similar yet faster solution with many of the same strengths.

However, most of the client's employees are more familiar with R which presents a strong case for using it to better incorporate it into their existing workflow.

Standalone Application

Another way we considered meeting our objective was by creating a standalone application that could run various data manipulation processes through a UI. The user would feed data to the application and choose the processes to be run on it. The UI would also allow for a significant amount of custom configuration for users to adjust as needed.

Strengths:
Doesn't require any coding to process data
Able to keep track of user preferences and state between sessions
Intuitive to use
Can easily save and reload models and results
Doesn't require installation of dependencies
Doesn't require user to use a particular language for other data manipulation

Weaknesses:
Difficult to feed results back into code
Unfamiliar concept for client's employees
Additional functionality requires building more UI
Less platform independent
Doesn't update along with packages automatically

We likely won't choose this approach mainly because our client wants the end product to be available to many users easily and without a high learning curve. This application would require users to learn an entirely new interface that doesn't even mesh well with their existing workflows.

Normally data is passed almost exclusively through code, so adding an application that has to import and export the data into the mix is somewhat awkward.

Browser Application

The final approach we considered was a browser application that could run all of the data manipulation processes remotely. The user could upload data and choose processes to be run on it. The application would provide a UI that could configure the processes as necessary. These processes would then be run on a server and the results sent back to the user.

Strengths:
All processes can be run on a server with above average processing power and memory
Results can be saved remotely and shared with other users
Doesn't require the user's computer to be available while running
Doesn't require coding to process data
Able to keep track of user preferences and state between sessions
Intuitive to use
Doesn't require installation of dependencies
Doesn't require user to use a particular language for other data manipulation

Weaknesses:
User must be online initially and to get the results
Big data must be uploaded and downloaded often
Difficult to feed results back into code
Unfamiliar concept for client's employees
Additional functionality requires building more UI
Requires a server to be accessed from

Similar to the previous approach, the browser application will likely not be chosen because of the learning curve and adjustment to workflow required. With the added complication of uploading and downloading the data, this approach could serious disrupt the flow of data without significant benefits.

2.2 Design Analysis

As the majority of the work on our project so far has been research, we don't currently have a chosen design to analyze at this time. The analysis for the several options we have can be found in the previous section.

As we move forward, we plan to get feedback from our client over our project ideas based on our research as soon as possible so that we can solidify our design for the project going forward. At that point we will begin constructing prototypes to begin the back and forth feedback loop with our client as we evolve the project.

3 Testing and Implementation

3.1 Interface Specifications

As our program is exclusively software based, there will be very minimal hardware interfacing necessary for this project. The solution we create will need to be manageable enough to run on a personal computer with no extra modifications.

In terms of software interfacing the solution we are creating will need to interface with Rstudio as that is what most of the data scientists at Principal use for writing scripts. It will also possibly need to interface with the database used by Principal as well as FactSet.

3.2 Hardware and software

Testing hardware will only require our own personal computers.

- Using our own personal computers allows us to regularly test our scripts.

Testing software will require an operating system that is capable of running R and Python scripts.

- Our own personal computers must be able to run Python and R in order to test scripts.
-

3.3 Functional Testing

We will perform unit testing through old scripts and programs Principal had. We will make sure the results our programs library outputs gives the exact same results as the old ones. We will integrate unit testing here with the outputs being the old programs inputs and outputs, and make sure our program provides the exact same results.

Once the the entire program is assembled together, acceptance testing will be conducted. We will first conduct the acceptance test ourselves to make sure all the requirements for the program have been met. Then during the usability test, acceptance testing will be carried out together, so the user can also confirm everything is in order.

When any general changes, or parts are added to the program, a set of regression tests will be carried out. These tests are carried out make sure that our program does not break when new parts are added to our program.

3.4 Non-Functional Testing

Testing for performance, security, usability, compatibility

Performance testing will be carried out by comparing our new program, compared to how the old program/scripts compared. We will running our program with the same configurations as the old

programs. These results will show our program performs compared to the old way Principal was doing things.

As our project will be used by the data science team, tests for usability will be extremely important. Once our prototypes are finished, we will hand our prototypes to Principal for user testing. We will observe in person how our program works under normal working conditions. Based on the results we have obtained from the users, changes will be made accordingly. This phase of testing will be done around the last phase of your project.

Compatibility testing will be done during the usability testing. Since the computers that will be primarily using our program are at Principal, the only way to conduct compatibility testing will be during the actual usability tests.

3.5 Process

Since we have not implemented any methods in our project as of yet, we do not have the specific ways we have tested the methods of our project as of yet.

3.6 Results

We haven't done any testing for our project yet, as we are still receiving feedback on our design ideas before moving on to implementation and testing.

4 Closing Material

4.1 Conclusion

So far our team has interviewed numerous Principal data scientists to get their feedback and input on what they specifically want, as well as examine and look into their redundant code. Our goal is create a unified application that can be used by all members of the team to reduce redundancies and wasteful time. We are currently waiting on feedback from Principal to decide which plan of action they deem is best.

4.2 References

No references at this time

4.3 Appendices

No appendices at this time