Quant Research Rapid Prototyping Library

Josiah Anderson | Doh Yun Kim | Gabe Klein | Drake Mossman | Jacob Richards | Nathan Schaffer
Project Contacts: Ben Harlander | Vishnu Vemuru
Project Advisor: Srikanta Tirthapura

1. Problem Statement

- The Global Equities team uses a process for developing stock performance prediction models called the Dynamic Risk Premium (DRP).
- This process currently does not meet Principal's standards for modularity, consistency, and transparency.
- Teams across the U.S. are working on the components of an improved "DRP 2.0", but no one is working on how those components will be integrated.

2. Proposed Solution

- Create a package of python functions and classes encompassing the entire DRP 2.0 process
- Fully implement functions that are standard for all runs of the pipeline
- Include abstract classes for modules that need to be implemented dynamically by the user
- Standardize interfaces within the pipeline as well as with the database

3. Requirements

Functional

- Consume raw stock level data
- Build models on factor portfolios, generate predictions and calculate model performance
- Each function in the library must be able to run independently
- The data created in this pipeline will be able to interface with a powerBI dashboard

Non-Functional

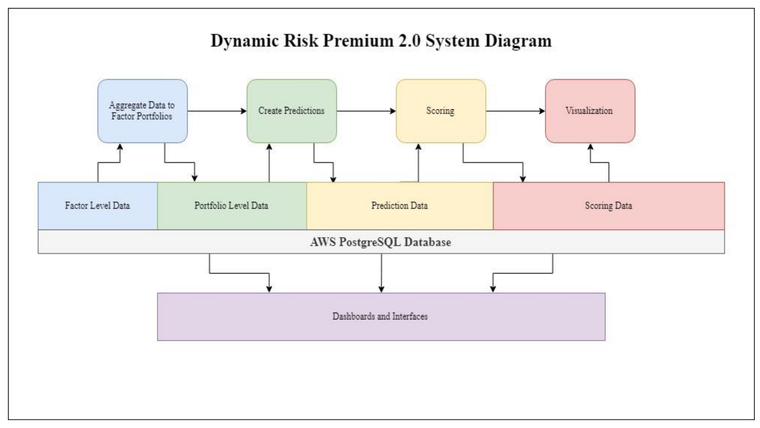
- (*Performance*) The pipeline will be able to handle millions of observations and hundreds of factors
- (*Maintainability*) The pipeline will have documented standards for each functions interface
- *(Usability)* The pipeline will be intuitive to a novice data scientist
- *(Accessibility)*The pipeline will require minimal formal coding experience to operate

4. Concept Diagram

Fully implemented:
- Aggregate Data
- Database

Abstract Classes:
- Create Predictions
- Scoring

Externally Implemented:
- Visualization
- Dashboards



5. System Architecture and Design

date	ticker	sales_g	mcap
08-21-2015	AAPL	25.930147	603119.94
	ABBV	8.972895	104983.71
	⋮	⋮	⋮
	ZTS	4.312786	22462.459
08-28-2015	AAPL	25.930147	646061.4
⋮	⋮	⋮	⋮
05-10-2019	ZTS	11.236173	45820.613

Aggregation Parameters

date	factor	⋯	aggregated return
08-21-2015	sales_g	⋯	2.509556
	mcap	⋯	0.066887
08-28-2015	sales_g	⋯	-7.881269
	mcap	⋯	-1.762554
⋮	⋮	⋯	⋮
05-10-2019	mcap	⋯	4.176405

Aggregation

- Aggregates raw stock level historical data to portfolios
- Provides ability to pre/post-process the raw data
- Completely implemented by us

date	sales_g predicted 5-year returns	mcap predicted 5-year returns
08-21-2015	5.323846	-1.653589
08-28-2015	4.327950	-0.288419
09-04-2015	3.926535	-0.693993
⋮	⋮	⋮
05-10-2019	7.048253	0.10582

Predictions

- Takes the factor portfolios and returns predicted returns
- Different predictions are returned depending on the different models being run

date	ticker	score
08-21-2015	AAPL	0.893346
	ABBV	0.506486
	⋮	⋮
	ZTS	0.676019
08-28-2015	AAPL	0.872793
⋮	⋮	⋮
05-10-2019	ZTS	0.660667

Scoring

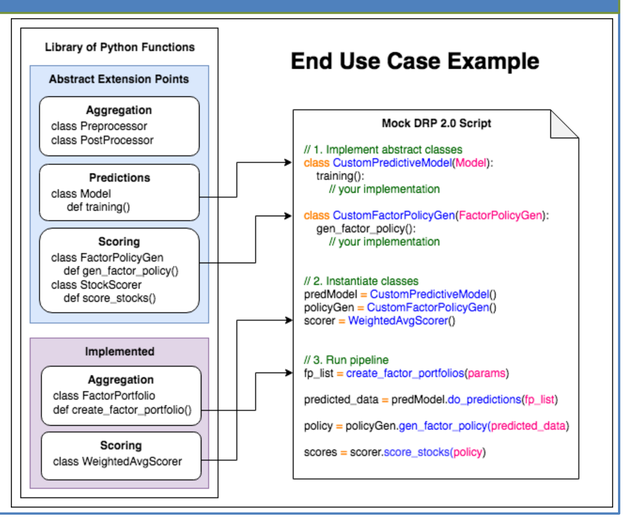
- Uses Portfolio-level predicted returns to find optimal factor weights (aka Factor Policies)
- Calculates stock-level scores for each date based on factor policy and historical data

6. Implementation

Our package contains classes and functions for every stage of DRP 2.0. Some classes are abstract for customization while we implemented logic for others.

Technologies

- Database:
  - PostgreSQL
  - AWS Server
- Pipeline
  - Python packages: numpy, pandas, psycopg2



7. Testing

Continuous Integration → Data Verification → System Testing → Use Case

Continuous Integration
- Gitlab tests running on a server acquired from ETG
- Tested for incompatible code being pushed to master

Data Verification
- Acquired data from Principal that had been aggregated using a trusted method
- Tested the output of our aggregation process against this data

System Testing
- Developed pseudo-instantiations of the pipeline
- Tested for data mismatch, incompatible interfaces

Use Case
- Provided code to potential users at Principal for feedback
- Tested for ease of use and intuitiveness of the pipeline

8. Conclusions

Current State

- There is a functional prototype of the entire DRP 2.0 pipeline from beginning to end.
- A database schema has been established and documented.
- Appropriate testing and documentation have been performed.

Aggregation Environment	Time Taken (seconds)	Time Taken (minutes)	Seconds Per Portfolio
No database interaction	1735.47	28.92	9.64
Save/Load, portfolios don't exist in database	2041.70	34.03	11.34
Save/Load, portfolios exist in the database	3401.00	56.68	18.89

Opportunities to Expand

- Although the aggregation step is fully implemented, there is still room to improve the speed of the process
- The format of the stock periods is not consistent with the standard Principal format
- This project is constantly evolving, so there are always new facets that could be fleshed out and implemented

Principal℠

IOWA STATE UNIVERSITY