Data Driven Method in Finance

Null Capital LLC

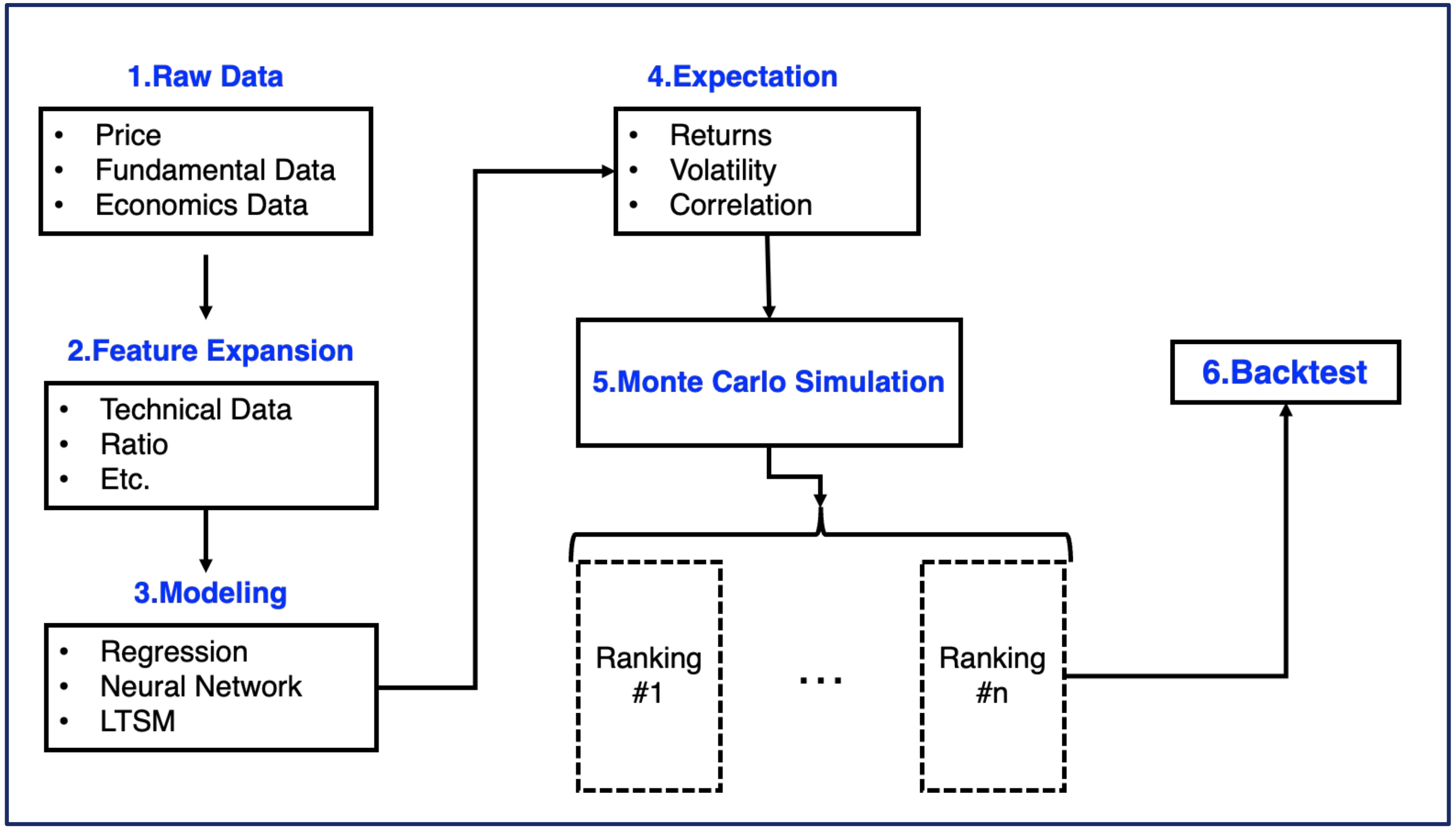
Jiayue Wu
Rachata Kongcheep
Vladislav Kim
Zixiao Yang

Agenda

- Our Approaches - 2
- Data - 1
- Model and Methods - 1
- Week to week flow of data - 1
- Results - 2
- Lesson Learned - 2

Approach (1/2) - Initial Approach

- Our goal is to build a **framework** that will ensure **robust performance** and **streamlined** our ideas into quantitative decision
- We then divided the framework construction into multiple modules. As shown below



Key component in Each Module

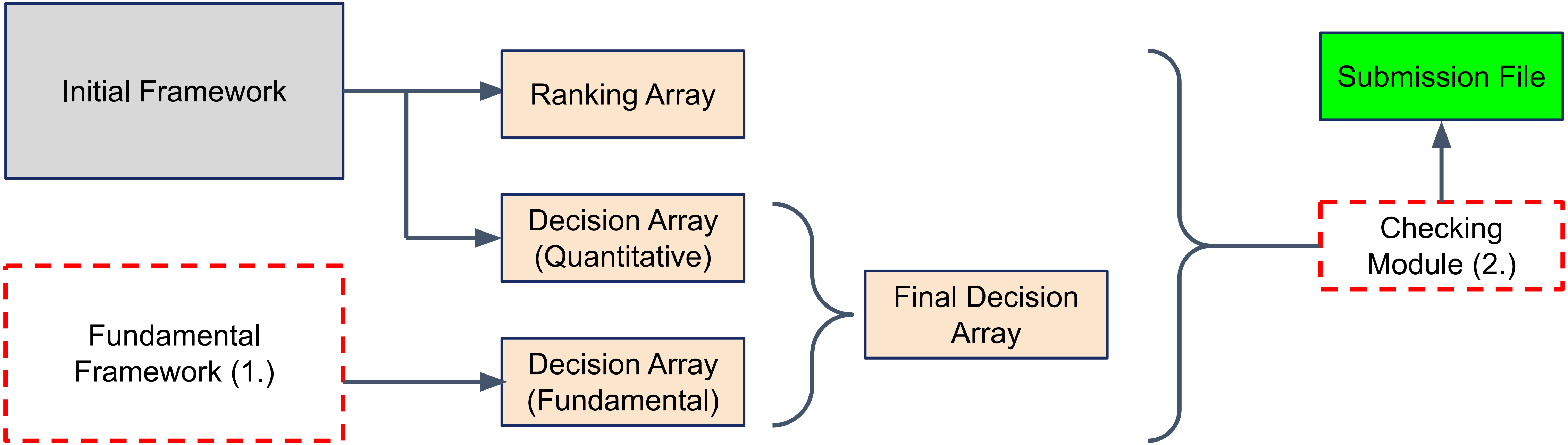
1. **Raw Data** - Get Data from from openbb, FRED, and yahoo finance
2. **Feature Expansion** - python Technical Analysis Library
3. **Modeling** - XGBoost
4. **Expectation** - manually adjusted through python and csv file
5. **Simulation** - Through python
6. **Backtest** - Through python

COLUMBIA | ENGINEERING
The Fu Foundation School of Engineering and Applied Science

Approach (2/2) - Implementation

As competition went on we made 2 improvement to the initial implementation

1. We add fundamental approach module to the framework - to express view that is harder to express quantitatively
2. Adding Checking Module at the end of the pipeline
3. We wrap each module into python object for easier implementation



Data

Our data came from 4 main sources each serving different dimensions of market view:

1. **Yahoo Finance**: for security prices and fundamental data
2. **FRED**: for macroeconomic data
3. **Estimize**: for trading earnings
4. **OpenBB**: for high level data ex. sentiment + insider trading

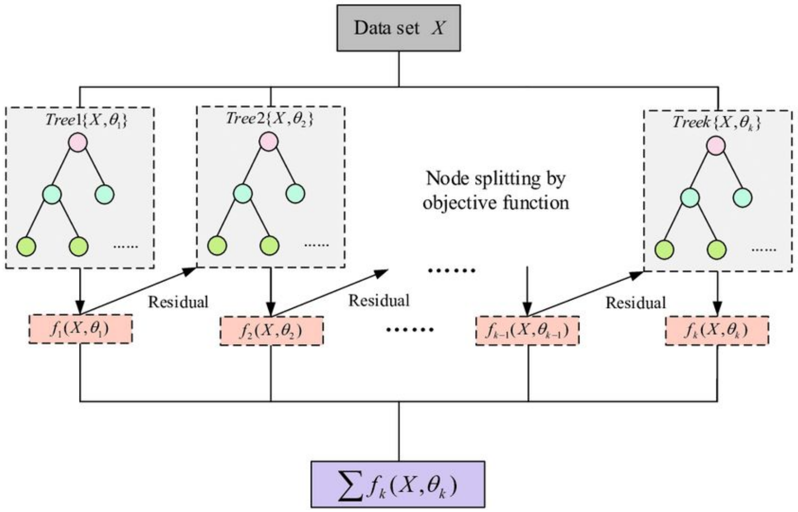
Overall, our size of features amounted to **427 features** for each asset

Model and Methods

- At first we starts modeling expected returns using various ranges of single models. These include:
  - Linear Models: OLS, Ridge, Lasso
  - **Support Vector Machine** - this being the one that give us the best results
  - Decision Trees
  - Neural Network Model
- **Then we moves toward ensemble methods:**
  - Bagging
  - Random Forest
  - AdaBoost
  - **XGBoost** - this being the best performing one

Thus, our main modelling engine is based on XGBoost model

XGBoost Model



Pros:

- Works well with large dataset
- Resilient and Robust Results

Cons:

- Black Box Nature - Hard to interpret
- Model is sensitive to outliers

Week to week flow

1. Import Related Module and Object

```
from Colab.code_base.data_utils import GetHistoricalData
from Colab.code_base.rank_simulator import RankSimulator

universe_df = pd.read_csv('data/universe_oct.csv')
```

2. Load Data via **GetHistoricalData** object

```
## Getting Historical Data
hd = GetHistoricalData(universe_df)
weekly_ret,daily_ret = hd.get_return_data()
hist_rank = hd.get_historical_ranking()
```

3. Read Loaded Data

```
## Read Fundamental Data
single_stock_data = pd.read_csv('data/single_stock_Data.csv',index_col=0)
x = single_stock_data.sort_values('TotalScore')
top_r_name = x[x['sector'] !='Communication Services'].dropna().index[:10].to_list()
bot_r_name = x[x['sector'] !='Communication Services'].dropna().index[-10:].to_list()


## Read Forecast Data
return_forecast = pd.read_csv('Colab/guy_mu_dec2.csv',index_col=0)
top_name = return_forecast.iloc[-1].fillna(0).sort_values(ascending=False).index.to_list()
bottom_name = return_forecast.iloc[-1].fillna(0).sort_values(ascending=True).index.to_list()
```

4. Allocate Weight

```
## Allocate Weights by Variance
ivp_list = top_name[:10] + bottom_name[:10]
cov_today = daily_ret[ivp_list].cov()
ivp_weight = getIVP(cov_today)
```

5. Simulate Ranking via **RankSimulator** object

```
## Get Rank Array by Simulation
rs = RankSimulator(return_forecast.iloc[-1].fillna(0)/252,daily_ret)
fore_mean = rs.simulate_rank()
```

6. Check and Save file

```
submission_file[['rank1', 'rank2', 'rank3', 'rank4','rank5']] = np.array(fore_mean)

submission_file['decision'] = np.array(weights)
checks(submission_file)

# submission_dir = dir_+'submissions/dueSep11/'
filename = 'NullCapitalLLC'+'__'+submission_time
submission_file.to_csv('submission/'+filename+'.csv')
print(filename)
```

Allocation

For Fundamental Approach: We Long/Short top/bottom 10 asset based on score calculated from these fundamentals

- return on assets
- total cash per share
- forward PE
- recommendation means

In order to get the weights for each name, we use solver to construct minimum variance portfolio from long and short names

COLUMBIA | ENGINEERING
The Fu Foundation School of Engineering and Applied Science

Results

- The results outperform the EW benchmark with 50% of the time (70% if week 7-8 is valid)
- Outperforming S&P 500 70% of the time (90% if week 7-8 is valid)

	group_name	weekly_rank_1	weekly_rank_2	weekly_rank_3	weekly_rank_4	weekly_rank_5	weekly_rank_6	weekly_rank_7	weekly_rank_8	weekly_rank_9	weekly_rank_10	overall_rank
0	LionQuant	9.00	3.50	1.50	2.00	5.50	5.00	3.00	8.00	7.00	1.50	4.600
1	EW	5.50	5.00	2.00	7.25	5.25	2.75	4.25	4.50	6.50	6.25	4.925
2	NullCapitalLLC	1.00	1.00	4.50	3.00	5.00	3.00	13.00	13.00	1.50	6.50	5.150

COLUMBIA | ENGINEERING
The Fu Foundation School of Engineering and Applied Science

Lesson Learned

- **Data is everything -**
  - We spent too much time experimenting with the modelling - we should've explore data more. Thing we should've explore more from data side includes:
    - Social Media Data
    - Quarterly and Annual Data
    - Data Cleaning Pipeline
    - Cross Sectional Analysis on Feature Importance for assets ex. which feature effects asset_i more; which feature can be excluded from asset_j
- **Macroeconomics events can be significant factor, especially for week-to-week basis;**
  - We tackle this by imposing manual view on top of portfolio (ex. Increase crypto short position in weeks with inflation print).
  - But there should be quantitative and robust ways to include macroeconomic events and views into decision making framework
- **Portfolio construction -**
  - We wish we had more time to explore a portfolio, which has a high correlation to the market when its rallying and market neutral or uncorrelated portfolio when the market is selling off.
- **Don't forget to check everything** - including the checking function

COLUMBIA | ENGINEERING
The Fu Foundation School of Engineering and Applied Science