

OSIG Research & Portfolio Analysis

Capstone Group 46

Project Members and Roles

Ashyan Rahavi: Machine learning development

Blake Cecil: Machine learning development

Joseph Noonan: Website development

Braeden Kuether: Project management, website design, workbook automation, project partner

What is OSIG?

- Oregon State Investment Group
- Manages approximately \$3.7 million across three portfolios.
- Analysts develop workbooks, reports and presentations on a company that they pitch to the group once a term.
- Portfolio managers adjust equity weights in their portfolio, buy/sell stocks, and coordinate with management on companies they would like pitched.



OREGON STATE
INVESTMENT
GROUP

Project Overview

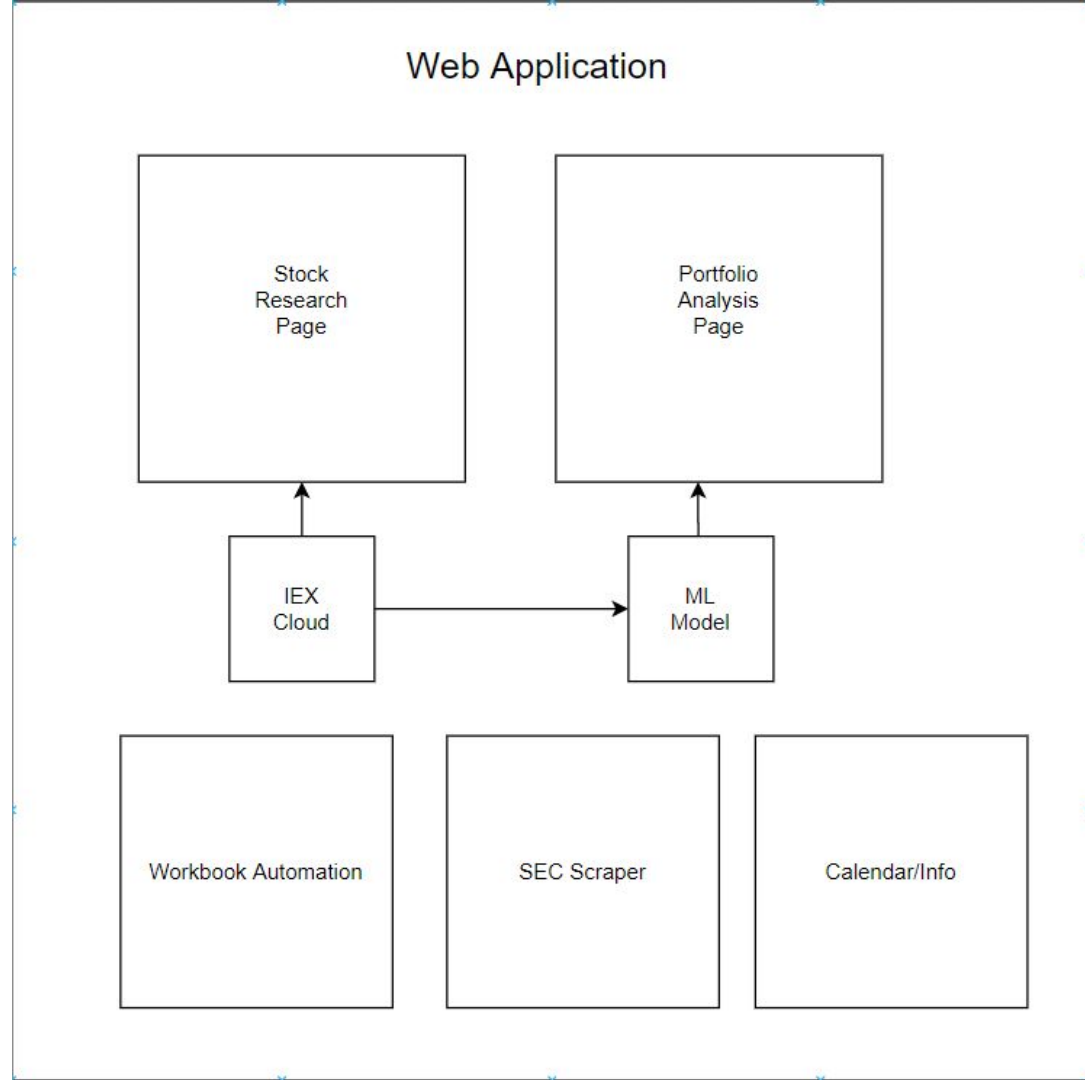
- Collect financial data from the IEX Cloud API
- Use machine learning to help manage portfolio decisions.
- Create research and portfolio analysis software for the Oregon State Investment Group.
- Create a central location for OSIG resources, including calendar/schedules, pitch voting results, important links and files, ect.
- Automate information gathering and model input in the OSIG workbook.

Overview Continued

- Web Application using react and django
- No build required on user end
- Website is limited for non OSIG members


Functional Diagrams

- Stock research page
- Workbook Automation
- SEC Scraper
- Calendar/Group Info
- Portfolio analysis page



Home Page


- OSIG management will handle the updating the results of the pitches on the calendar
- OSIG will also handle putting the upcoming and past pitches on the calendar on the home page
- The calendar is currently functioning as intended
- Resources are used to help complete the workbook, such as the SEC EDGAR website, the workbook template, country risk premiums, and daily treasury yields
- Resources also contain links to the group Box and website

Home Stock Research  Portfolio Analysis About

Oregon State Investment Group

Today May 2021 Print Week Month Agenda

Sun	Mon	Tue	Wed	Thu	Fri	Sat
25	26	27	28	29	30	May 1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	Jun 1	2	3	4	5

Events shown in time zone: Pacific Time - Los Angeles 

Resources

OSIG Website	link
Box Home	link
Box Minutes	link
Box Pitches	link
Box Portfolios	link
SEC EDGAR	link
Workbook Template	link
Daily Treasury Yields	link
Country Risk Premiums	link

Data provided by IEX Cloud

Stock Research Page

- Data provided includes income statement, balance sheet, cash flows, and summary statistics
- Any company covered in the IEX Cloud API can be displayed
- Can enter competitors and download relevant statements in a zip file
- Workbook can use the zip file to complete certain sections

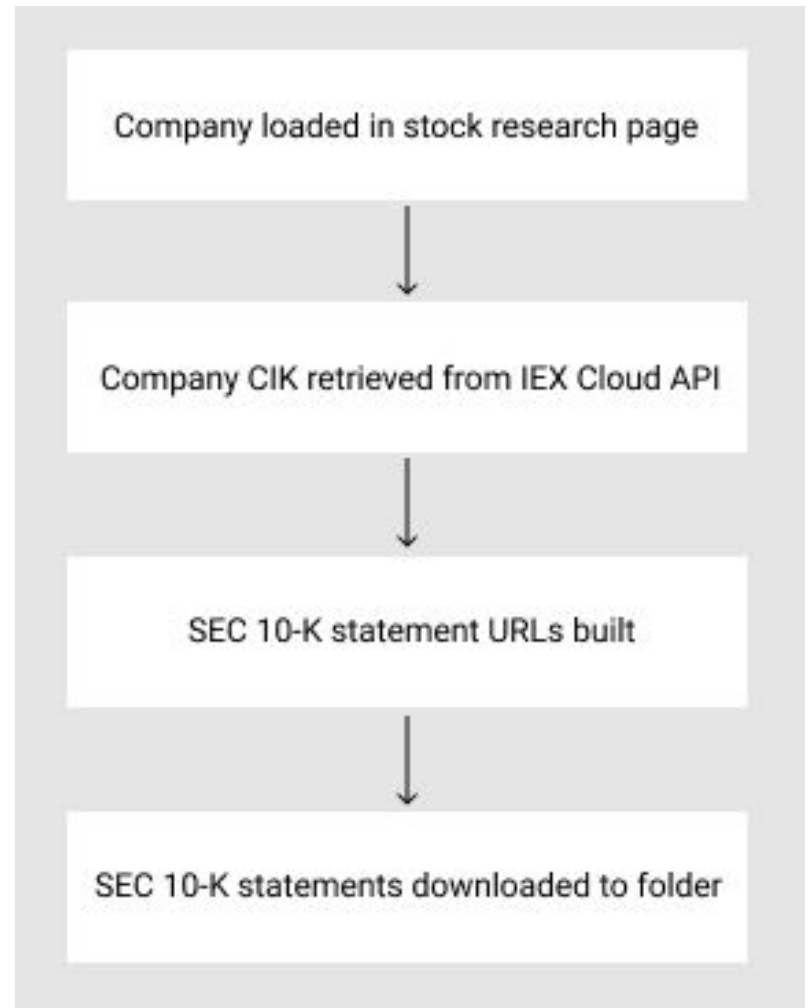
The screenshot displays the IEX Cloud Stock Research interface. At the top, there is a navigation bar with links for Home, Stock Research, INVESTMENT (with a logo), Portfolio Analysis, and About. The main content is divided into two columns. The left column contains two sections: 'Data' and 'Export Workbook'. The 'Data' section has input fields for 'Stock Ticker' (MSFT) and 'Form Choice' (Income Statement), with a 'Submit' button below. The 'Export Workbook' section has input fields for 'Stock Ticker' and four 'Competator' fields, with an 'Export' button below. The right column is titled 'Income Statement' and shows 'Year: 2021'. It contains a table with financial metrics and their values.

Income Statement	
Year: 2021	
costOfRevenue	46574655973
currency	USD
ebit	54001154318
filingType	10-K
fiscalDate	2020-06-21
fiscalQuarter	0
fiscalYear	2021
grossProfit	100154236768
incomeTax	8932483947
interestIncome	2597404890
minorityInterest	0
netIncome	45163432830
netIncomeBasic	45670483058
operatingExpense	90759323073
operatingIncome	53398404842
otherIncomeExpenseNet	0

Data provided by IEX Cloud

Workbook Automation

- OSIG uses SEC 10-K statements to build workbooks
- This simple script retrieves those statements in excel format




Current Architecture - High Level Process Flow

1. Investor picks a set of assets for a new portfolio they want to create.
2. Stock Features are grabbed for the past 5 years.
3. Neural Net trains on the history of these stock prices and optimizes based on the Sharpe Ratio.
4. Recommended weights for each asset is returned once the model is finished training.

Portfolio Analysis Page

- Create portfolios with tickers from IEX cloud
- Analyze them on the back end and send recommendations, stats visualizations etc


Home Stock Research  Portfolio Analysis About

Portfolio Analysis

Title:

Previous Runs

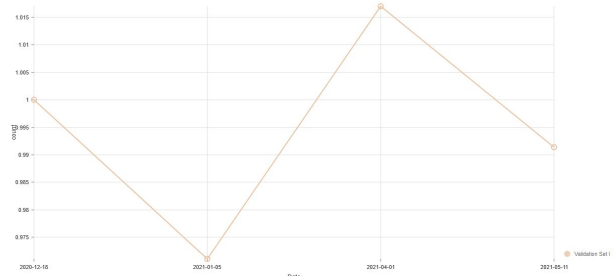
TITLE	DATE	ID
Demo Sample Run	2021-05-12 23	
msl.LC.test_1000	2021-05-12 22	
msl.LC.test	2021-05-12 21	
msl.LC.test	2021-05-09 20	
msl.LC.test	2021-05-09 19	
msl.LC.test	2021-05-09 18	
msl.LC.test	2021-05-09 17	
msl.LC.test	2021-05-09 16	
msl.LC.test	2021-05-09 15	
msl.LC.test	2021-05-08 14	
test	2021-05-08 13	
msl.test - no ex 2	2021-05-05 12	
msl.test - no ex 1	2021-05-05 11	
test6	2021-05-02 10	
test	2021-05-02 9	
L.C.2	2021-04-28 8	
L.C.	2021-04-28 7	
large_cap_sample_3	2021-04-20 6	
tester_2	2021-04-20 5	
tester_2	2021-04-20 4	
tester	2021-04-20 3	
large_cap_sample_2	2021-04-20 2	
large_cap_sample	2021-04-15 1	

Home Stock Research  Portfolio Analysis About

Recommended Weighting

pypl: 0.2250796253719846 € 0.2573243135703298 csco: 0.5175960610576855

Simulated Returns With ML Model



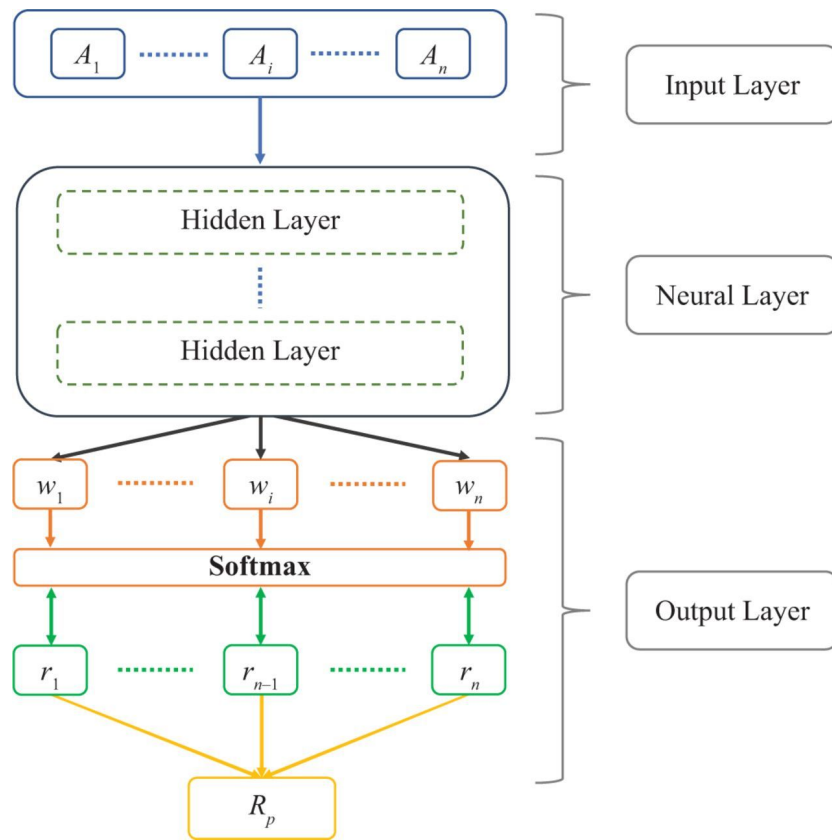
Date	Return
2020-12-15	1.000
2021-01-05	0.975
2021-04-01	1.015
2021-05-15	0.995

Cumalitive Returns With Inpupped Weights

Functional Diagram

- LSTM network.
- Sharpe ratio as loss function.

$$R_{p,t} = \sum_{i=1}^n w_{i,t-1} r_{i,t}$$
$$E(R_{p,t}) = \frac{1}{T} \sum_{i=1}^T R_{p,t}$$
$$L = \frac{E(R_{p,t})}{\sqrt{\text{var}(E(R_{p,t}))}}$$



ML Model Details

```
def sharpe_loss(weights, returns):
    # weights batch * time * assets
    # returns batch * time * assets
    #print(weights.shape, returns.shape)

    #row wise dot product
    R = torch.sum(weights*returns,dim=-1)
    ER = torch.mean(R,1)
    STD = torch.std(R,1)
    ratio = torch.sum(ER/(STD+1e-6))
    return -ratio
```

```
class Net(nn.Module):
    def __init__(self,NUM_FEATURES,NUM_ASSETS,TIME_PERIOD_LENGTH):
        super(Net, self).__init__()
        self.time = TIME_PERIOD_LENGTH
        self.input = nn.LSTM(NUM_FEATURES, 64, 1, batch_first = True)
        self.lin = nn.Linear(64,NUM_ASSETS)
        self.soft_out = nn.Softmax(dim=2)
```

Future Work

- Design more robust ML model
- More analysis tools (price prediction, asset selection, etc)
- More data collection
- Automate more tasks