Producing an optimum stock portfolio using Web Scraping, Information Extraction and Genetic Algorithms

Ben Winter

Stock portfolio

- A stock portfolio is a collection of stocks from different companies, of which an investor would place a portion of their funds into.
- Usually these are created by financial advisors, stock brokers or wealth management teams.
- Not only do they cost money to create, the team who created your portfolio will typically take 1-10% commission on any profit that portfolio brings in.
- Portfolios are used in order to spread the risk of an investment and try to maximise potential returns.

How are they created?

- To create a portfolio a team/person will research current trends, read news articles regarding potential investable companies, and of course read financial information such as historical data on companies.
- Portfolios are created with the notion of having a low risk, with the highest possible return. However, this may vary for different investors, some people might want to take a risk in order to get a higher return etc.
- Once a portfolio is created, the investor is told and called in for a meeting to see if the portfolio is acceptable.

The problem with this

- Not only do some people not have the money to afford commission fees, it also requires them to go outside, find a financial advisor or someone of a similar profession and understand what the financial advisor is talking about. For many people, the language of the financial world will be jargon.
- There is also the problem of an advisor putting their own bias into your portfolio, whether this is a conscious or subconscious bias.

My solution

- My solution includes wrapping the research, assessment and creation of the portfolio into a single executable that the investor themselves can run.
- This is done by firstly using web scraping techniques to scrape news articles from a ‘reputable’ news source, pooling the most “in favour” companies.
- Then scraping the last years worth of historical stock price of said companies.
- The program then uses the mean variance model in conjunction with genetic algorithms to find the most optimum portfolio.

Web Scraper 1.0

- This optional part of the program needs to source news articles, run for a user-defined period of time (until cancellation of program) and copy each news article from a business news site.
- To reduce the storage needed for this web scraper, some information extraction tools are used. Namely: the removal of stopwords, the removal of capitalization and the removal of punctuation.
- These articles are then saved in a .txt file, and checked to see if a company name is amongst the article. If a company is found it is then checked against a well known Sentiment Analysis file called AFINN-111, which looks at each word in the article and ranks each word based on its positivity.

```
abandon -2
abandoned       -2
abandons        -2
abducted        -2
abduction       -2
abductions      -2
abhor   -3
abhorred        -3
abhorrent       -3
abhors  -3
abilities       2
ability 2
aboard  1
absentee        -1
absentees       -1
absolve 2
absolved        2
absolves        2
absolving       2
absorbed        1
abuse   -3
abused  -3
abuses  -3
abusive -3
accept  1
accepted        1
accepting       1
accepts 1
accident        -2
accidental      -2
accidentally    -2
accidents       -2
accomplish      2
accomplished    2
accomplishes    2
accusation      -2
accusations     -2
accuse  -2
accused -2
accuses -2
accusing        -2
```

Web Scraper 2.0

- After 5 companies have been found, a second web scraper runs to check and save those companies stock price at closing for every month of the last year.
- These prices are then stored in 5 different arrays.

The Mean-variance model

- The Mean-Variance model is used to compare the risk and returns of a portfolio.
- A typical portfolio for this program will look like

Companies: A B C D E

[0.32, 0.01, 0.29, 0.06, 0.32] = 1

- Each number represents a percentage of an investors budget that should be used to place stocks on company X

Return of Portfolio: The expected returns of a portfolio can be found by the the weights multiplied by the mean returns of each individual. Expected returns therefore has an equation of:

$$R_\rho = (\omega_\iota \ \mu_\iota) + (\dots * \dots) + (\omega_\nu \ \mu_\nu)$$

where:

- R_ρ is a given portfolio
- ω is a given weight
- ι is a given company
- μ is a mean percentage return
- ν is the amount of companies

Risk of portfolio: The calculated risk of a portfolio as calculated using the source [5].

$$\begin{aligned} r_\rho = (((\omega_\iota\sigma_\iota)^2 + (\omega_\jmath\sigma_\jmath)^2 + (\omega_\kappa\sigma_\kappa)^2 + (\omega_\ell\sigma_\ell)^2 + (\omega_m\sigma_m)^2)+ \\ ((\omega_\iota\omega_\jmath\sigma_\iota\sigma_\jmath) + (\omega_\iota\omega_\kappa\sigma_\iota\sigma_\kappa) + (\omega_\iota\omega_\ell\sigma_\iota\sigma_\ell) + (\omega_\iota\omega_m\sigma_\iota\sigma_m) + (\omega_\jmath\omega_\kappa\sigma_\jmath\sigma_\kappa)+ \\ (\omega_\jmath\omega_\ell\sigma_\jmath\sigma_\ell) + (\omega_\jmath\omega_m\sigma_\jmath\sigma_m) + (\omega_\kappa\omega_\ell\sigma_\kappa\sigma_\ell + (\omega_\kappa\omega_m\sigma_\kappa\sigma_m) + (\omega_\ell\omega_m\sigma_\ell\sigma_m)) \end{aligned} \quad (5.1)$$

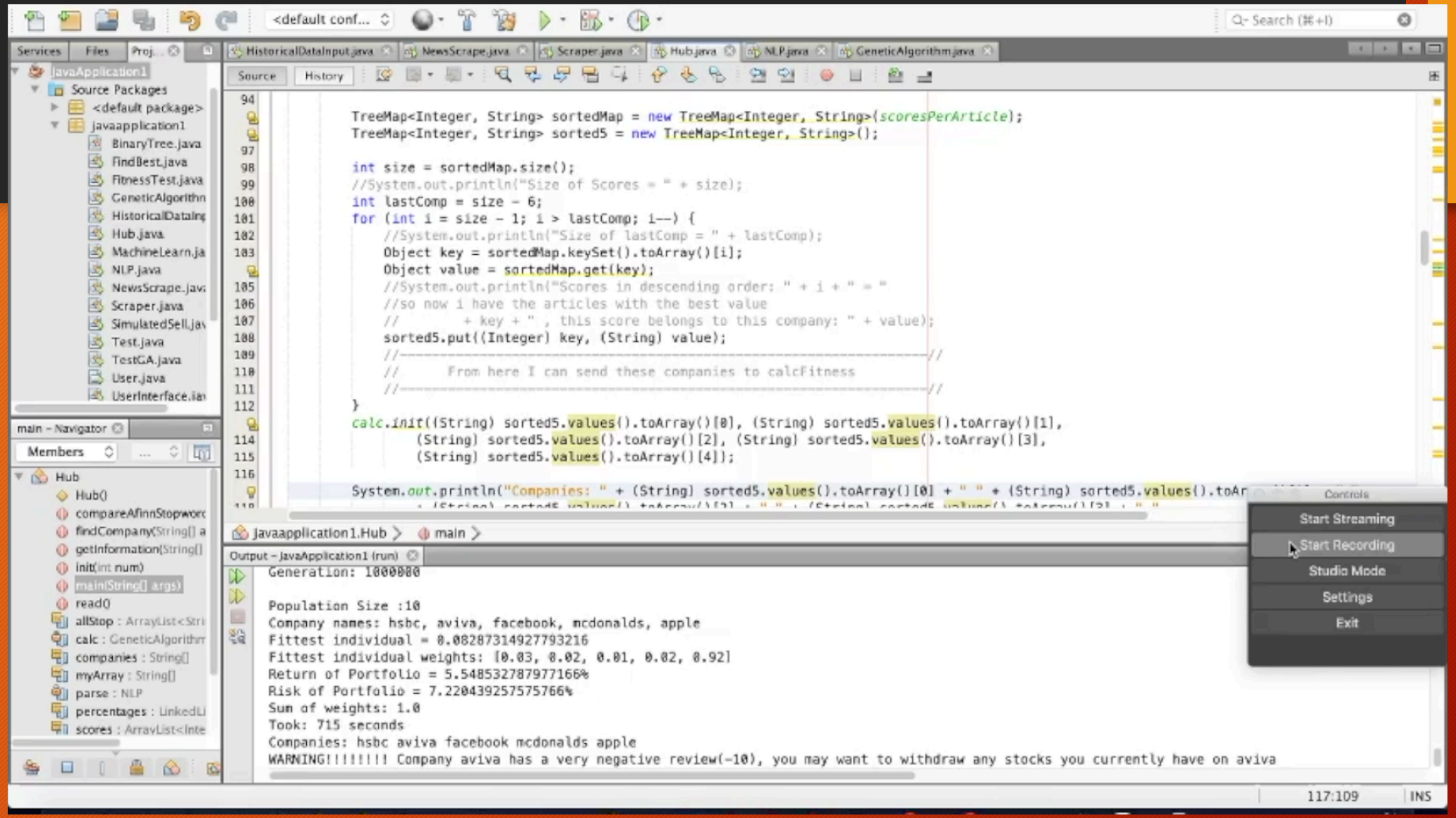
$$r_\rho = \sqrt{r_\rho} \quad (5.2)$$

where:

- r_ρ is the risk of the portfolio
- ω is a given weight
- σ is a standard deviation of the company
- ι is company ι
- $\jmath$ is company $\jmath$
- κ is company κ
- ℓ is company ℓ
- m is company m

Genetic Algorithms

- A random population of portfolios are initially created.
- The fitness of each portfolio is checked by using the Mean-Variance model.
- Genetic modifiers such as crossover, mutation and elitist selection are used.
- After a certain amount of generations an optimum portfolio will be created. Which will be the portfolio with the lowest risk for the highest return.


TreeMap<Integer, String> sortedMap = new TreeMap<Integer, String>(scoresPerArticle);
TreeMap<Integer, String> sorted5 = new TreeMap<Integer, String>();
int size = sortedMap.size();
//System.out.println("Size of Scores = " + size);
int lastComp = size - 6;
for (int i = size - 1; i > lastComp; i--) {
//System.out.println("Size of lastComp = " + lastComp);
Object key = sortedMap.keySet().toArray()[i];
Object value = sortedMap.get(key);
//System.out.println("Scores in descending order: " + i + " = "
//so now i have the articles with the best value
// + key + " , this score belongs to this company: " + value);
sorted5.put((Integer) key, (String) value);
//From here I can send these companies to calcFitness
}
calc.init((String) sorted5.values().toArray()[0], (String) sorted5.values().toArray()[1],
(String) sorted5.values().toArray()[2], (String) sorted5.values().toArray()[3],
(String) sorted5.values().toArray()[4]);
System.out.println("Companies: " + (String) sorted5.values().toArray()[0] + " " + (String) sorted5.values().toAr
Controls
Start Streaming
Start Recording
Studio Mode
Settings
Exit
Generation: 1000000
Population Size :10
Company names: hsbc, aviva, facebook, mcdonalds, apple
Fittest individual = 0.08287314927793216
Fittest individual weights: [0.03, 0.02, 0.01, 0.02, 0.92]
Return of Portfolio = 5.548532787977166%
Risk of Portfolio = 7.220439257575766%
Sum of weights: 1.0
Took: 715 seconds
Companies: hsbc aviva facebook mcdonalds apple
WARNING!!!!!!!! Company aviva has a very negative review(-10), you may want to withdraw any stocks you currently have on aviva
117:109
INS

```
Company names: nestle, volkswagen, facebook, shell, starbucks
Fittest individual = 0.07549872305283833
Fittest individual weights: [0.02, 0.02, 0.02, 0.02, 0.92]
Return of Portfolio = 4.669011965655982%
Risk of Portfolio = 5.794507831974 02%
```

Thank you, any questions?