**Requires pandas\_datareader package to run**

* pip install pandas\_datareader

<https://pandas-datareader.readthedocs.io/en/latest/>

The GA python script attached recalculates the population if it finds an empty list while processing the population. This is because such a value returns ‘inf’ while calculating fitness for the population. This is done in a while loop, twice, once for the initial set of the population, and the other for the crossover generation.

There is also a hacky ‘if’ statement on line 210, because if I delete the indices in ascending order, then that will result into resetting of indices in the list, and I’ll just end up deleting a Fit individual.

The crossover is performed by randomly taking from a set of ‘genes’ generated by ‘generate\_population’ in the initial population, and then randomly selecting values from each of the rows, to form two rows that will be used for crossovers. While processing the population, mutation occurs if the flag is set, and this results into a slight bump in the values of the conditions against which the price is being compared with.

To predict the future values, a simple Linear Regression was performed, with half the output of the GA algo used as a training set.

**IMPROVEMENT:**

1. There is an iteration loop. For now, only 1 iteration is taken, that is, after unfit individuals are removed, a new set of individuals are generated by crossover and mutation, and added to the population. The fitness of this new population is calculated and returned. If the number of iterations were greater (which is just a number change in the script), then, more unfit individuals would be discarded, and new individuals would be generated by crossover and mutation
2. The number of unfit individuals which are discarded per iteration are 2. This could be improved by change in code, but has not been incorporated
3. The calculation was made on daily returns, since in stock prediction, returns are a more tangible figure than static prices

**Note to Professor:**

This Mini-Project was slightly tricky. The theoretical part didn’t make it easy to understand. Luckily, the forums with students giving code hints made things much clearer. Would like it if you would give students some code hints like the previous ones, since it helps those from Non-Programming backgrounds.