

Amna Irfan
CS 441- Hw3
11/21/2019

Monte-Carlo Stock Prediction using Apache Spark

Introduction:

The spark frameworks allows the parallel processing of datasets in which records are independent of each other. Unlike Hadoop MapReduce, Apache Spark support time-series data that depends on each other as well. In this project, we build a stock-prediction Monte-carlo simulator that takes a date range and starting investment (in USD) as it's input and in return, it generates stock-related decisions a user can take each day. These decisions will be of either buying a stock or selling it on a particular day.

Dataset and preprocessing: In order to run the simulation, we first needed a stock timeseries data. Instead of picking random stocks, we choose 500 stocks from an index fund *iShares Core S&P 500 ETF (IVV)*. We then created a [python script](#) to request the time series data of each of the 500 stocks using the Alpha-advantage API.

A small sample of the data for the AAPL stock from Alpha-advantage.

Timestamp	Open	High	Low	Close
2019-11-15	263.680	265.77	263.76	265.76
2019-11-14	263.750	264.88	262.66	262.64

Once we downloaded all the data from Alpha-advantage, we ran another [python script](#) to reformat and sample the data. In order to keep our simulation simple, we discarded all records that were older than a year. We also made sure that none of the stocks in our data had any missing day. In our data, we only kept the symbol, the timestamp and the closing value of the stock. All other properties of the time-series data was discarded.

Functionality: Our simulator creates multiple simulations of decisions that a user can take regarding buying and selling stocks for each day. The results of a simulation is something like this for a single day

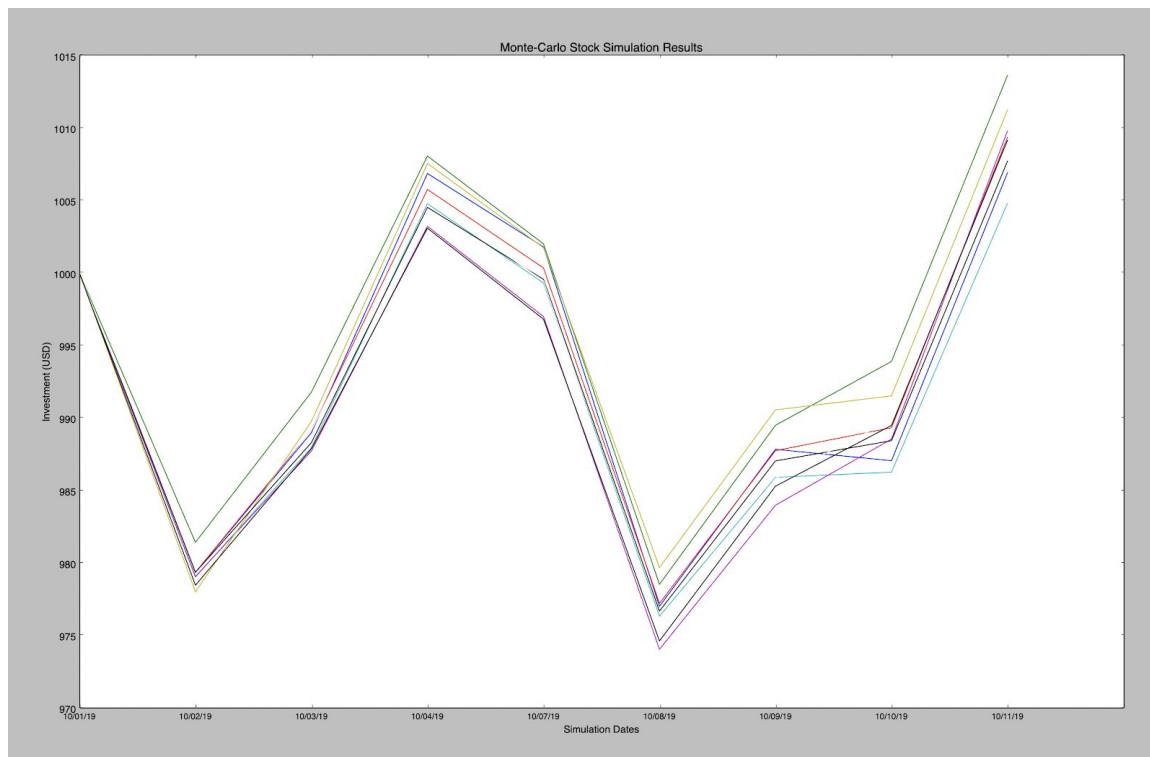
Timestamp	Symbol	Close	Share
2019-11-15	AAPL	264	0.5
2019-11-15	WEC	32.64	3.6
2019-11-15	WMB	15.6	2.0

The *Share* column indicates the number of stocks the investor has purchased. For example, this table shows the investor has purchased half a stock of AAPL which means he has \$132 invested in that stock. An entry could either mean that the investor bought the stock on that day or just didn't sell it from before and kept it in their stock profile. We follow the following steps in order to make buy/sell decisions on stocks

1. We use Spark dataframes to load the stock time series data.
2. We remove weekends from the date range and holidays as well that the stock market observes.
3. A sample of N stocks are chosen from our pool of 500 stocks
4. For each stock, we choose random N time-series data and calculate the average loss for that stock. The loss is calculated by getting the difference between the closing value of two random dates of the specified stock. A negative average loss value indicates that on average the stock value increases for that stock.
5. For each stock, we calculate the 0.25, 0.5, 0.75 and 0.95 percentile on closing values.
6. We use 3 and 4 in our buy and sell policy
7. To predict the stock profile on the first day, we choose random N stocks and invest a weighted amount on them based on their average loss. We call this the “**Loss Based Buy Policy**”. For example, if we randomly choose to invest in two stocks and one has a higher average loss than the other, we will invest a smaller percentage of our money on that stock and a higher percentage on the other.
8. To predict the stock profile for the next days (other than the second) where we have a previous stock profile, we do the following
 - a. First, we sell stocks based on the “**Percentile Based Sell Policy**”. In this policy, we sell a stock if it's value is more than it's 75 percentile. The intuition behind this is that if a stock value has reached it's top 75%, it's best to sell it as there are chances it will go down in near future.

- b. Once stocks are sold, we calculate the new available fund amount so that we can buy new stocks. We again use the loss-based buy policy to make that decision.
- c. We then merge the stocks that we did not sell and the stocks that we just bought into a single profile which is then used in the next iteration of stock profile prediction/decision.

Results: We ran 10 simulations with a sample of 100 stocks for the date range *10/01/19* -*10/12/19* with the starting investment of \$1000. Below is a chart that shows the trend of total money made on each day based on our policies for all 10 simulations.



Here we can see that the investment amount does not change much. This indicates that we could have used better policies or more data to calculate the random average losses. Also notice that all 10 the simulations follow a similar trend. Our results can be viewed here

<https://s3.console.aws.amazon.com/s3/buckets/org.amnairfan.cs441.hw2/results/?region=us-east-1&tab=overview>

The daily investment money is in the result folder of each simulation folder. The other folders are the daily stock profiles with the stock names, the closing amount on that day and the investor's share on that stock for that day.

Google Dataproc:

Google Cloud Platform

My First Project

data

Clusters

Jobs

Workflows

Create a cluster

Name ⓘ

aminalfan-monte-carlo-stock-sim

Region ⓘ

us-central1

Zone ⓘ

us-central1-a

Cluster mode ⓘ

Standard (1 master, N workers)

Master node

Contains the YARN Resource Manager, HDFS NameNode, and all job drivers

Machine type ⓘ

4 vCPUs

15 GB memory

Customize

Upgrade your account to create instances with up to 96 cores

Primary disk size (minimum 15 GB) ⓘ

500

GB

Primary disk type ⓘ

Standard persistent disk

Worker nodes

Each contains a YARN NodeManager and a HDFS DataNode.

The HDFS replication factor is 2.

Machine type ⓘ

4 vCPUs

15 GB memory

Customize

Upgrade your account to create instances with up to 96 cores

Primary disk size (minimum 15 GB) ⓘ

500

GB

Primary disk type ⓘ

Standard persistent disk

Nodes (minimum 2) ⓘ

7

0

x 375 GB

YARN cores ⓘ

8

YARN memory ⓘ

24 GB

Autoscaling policy ⓘ (Optional)

☐ Enable autoscaling on the cluster.

This project does not currently have any applicable policy to enable autoscaling in this region. [Learn how to create autoscaling policy.](#)

Component gateway

☐ Enable access to the web interfaces of default and selected optional components on the cluster. [Learn more](#)

Free trial status: \$300.00 credit and 365 days remaining - with a full account, you'll get unlimited access to all of Google Cloud Platform.

Google Cloud Platform My First Project


Dataproc Clusters

CREATE CLUSTER REFRESH DELETE REGIONS Choose the Cloud Dataproc regions to list

SHOW INFO PANEL

Search clusters, press Enter

Name	Region	Zone	Total worker nodes	Scheduled deletion	Cloud Storage staging bucket	Created	Status
ammarfan-monte-carlo-stock-sim	us-central1	us-central1-a	3	Off	dataproc-edbe2a04-d56c-4219-83b1-a6f751fabce0-us-central1	Nov 26, 2019, 9:29:49 PM	Provisioning



Google Cloud Platform My First Project

Cluster details

amniaifan-monte-carlo-stock-sim

For PD-Standard without local SSDs, we strongly recommend provisioning 1TB or larger to ensure consistently high I/O performance. See <https://cloud.google.com/compute/docs/disks/performance> for information on disk I/O performance.

Monitoring Jobs VM instances Configuration Web interfaces

Name	Role
amniaifan-monte-carlo-stock-sim-m	Master
amniaifan-monte-carlo-stock-sim-w-0	Worker
amniaifan-monte-carlo-stock-sim-w-1	Worker
amniaifan-monte-carlo-stock-sim-w-2	Worker

SSH

Equivalent REST