

Bar Chart Application On Google Cloud App Engine Part 1



Dev Thakkar · [Follow](#)

4 min read · Nov 28, 2021



Scenario: Review four weeks of multiple API transaction calls to identify slowdown in response times

Tech stack: Google Cloud Storage, App Engine, Python, Visual Studio Code (VS Code)

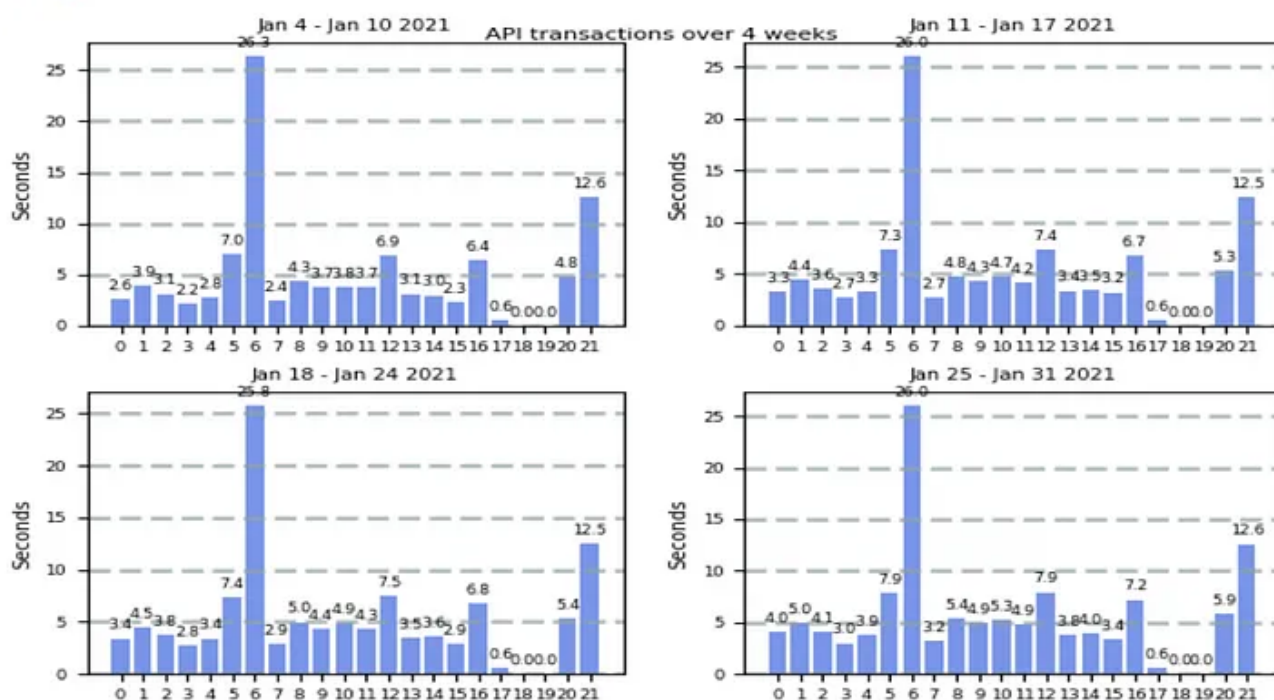
Requirements:

- (1) Store metrics csv files in cloud storage
- (2) Use matplotlib to generate transaction response times bar charts
- (3) Deploy python application to App Engine

Outcome:

update graph

Plot as a pag



Step 1: Prerequisites

- Setup [Google Cloud](#) (GCP) account
- Download [VS Code](#). Install extensions for Python

Step 2: Create new GCP Project and setup Cloud Storage Bucket

- Project Name — responsegraph; Project ID —responsegraph
- Link to setting up project for [App Engine](#)
- Create new bucket “graphbucket_test” (link to [Cloud Storage Quickstart](#))

← Bucket details

graphbucket_test

Location	Storage class	Public access	Protection
us (multiple regions in United States)	Standard	Not public	None

OBJECTS CONFIGURATION PERMISSIONS PROTECTION LIFECYCLE

Buckets > graphbucket_test

[UPLOAD FILES](#)
 [UPLOAD FOLDER](#)
 [CREATE FOLDER](#)
 [MANAGE HOLDS](#)
 [DOWNLOAD](#)
 [DELETE](#)

Filter by name prefix only ▾ Filter Filter objects and folders

<input type="checkbox"/>	Name	Size	Type	Created ?	Storage class	Last modified	Public access ?
<input type="checkbox"/>	Jan 11 - Jan 17 2021.csv	1 MB	application/vnd.ms-excel	Nov 23, 2021, 6:05:55 AM	Standard	Nov 23, 2021, 6:05:55 AM	Not public
<input type="checkbox"/>	Jan 18 - Jan 24 2021.csv	1 MB	application/vnd.ms-excel	Nov 23, 2021, 6:05:55 AM	Standard	Nov 23, 2021, 6:05:55 AM	Not public
<input type="checkbox"/>	Jan 25 - Jan 31 2021.csv	1 MB	application/vnd.ms-excel	Nov 23, 2021, 6:05:57 AM	Standard	Nov 23, 2021, 6:05:57 AM	Not public
<input type="checkbox"/>	Jan 4 - Jan 10 2021.csv	828.1 KB	application/vnd.ms-excel	Nov 23, 2021, 6:05:55 AM	Standard	Nov 23, 2021, 6:05:55 AM	Not public

- Upload metrics csv files to *graphbucket_test*
- Review sample CSV file data from four files

```

Jan 4 - Jan 10 2021.csv
1  server,process,responsetimeorcount
2  tomcat2,details|Responses,20
3  tomcat1,casualty|Responses,10
4  tomcat2,search|Response Time,21
5  tomcat1,summary|Response Time,2
6  tomcat2,casualty|Response Time,4798
7  tomcat1,details|Response Time,721
8  tomcat2,results|Response Time,553
9  tomcat2,info|Responses,3
10 tomcat2,results|Responses,4
  
```

subset of 26,000+ rows. Columns: (a) server (b) process and (c) responsetimeorcount

```

Jan 11 - Jan 17 2021.csv
1  server,process,responsetimeorcount
2  tomcat1,details|Response Time,3878
3  tomcat2,summary|Response Time,7
4  tomcat2,documentation|Response Time,4237
5  tomcat1,summation|Responses,3
6  tomcat1,details|Response Time,5890
7  tomcat1,conversion|Responses,5
8  tomcat2,details|Responses,8
9  tomcat2,report|Responses,3
10 tomcat2,search|Response Time,11
  
```

subset of 26,000+ rows. Columns: (a) server (b) process and (c) responsetimeorcount

Jan 18 - Jan 24 2021.csv		
1	server,process,responsetimeorcount	
2	tomcat2,search Response Time,3	
3	tomcat1,information Response Time,7445	
4	tomcat2,results Response Time,554	
5	tomcat2,results Responses,8	
6	tomcat2,assignment Response Time,7518	
7	tomcat1,info Responses,14	
8	tomcat1,summary Responses,4	
9	tomcat1,summary Response Time,10	
10	tomcat1,info Response Time,7848	

subset of 26,000+ rows. Columns: (a) server (b) process and (c) responsetimeorcount

Jan 25 - Jan 31 2021.csv		
1	server,process,responsetimeorcount	
2	tomcat2,info Responses,12	
3	tomcat2,details Response Time,5	
4	tomcat2,details Responses,7	
5	tomcat1,info Responses,3	
6	tomcat2,results Responses,9	
7	tomcat2,search Responses,3	
8	tomcat1,summary Response Time,6	
9	tomcat1,search Responses,5	
10	tomcat1,summary Responses,7	

subset of 26,000+ rows. Columns: (a) server (b) process and (c) responsetimeorcount

Step 3: Setup local development environment

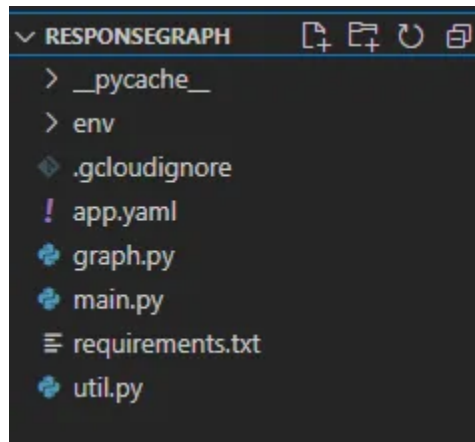
- Install Cloud SDK to your local (skip if already installed)
- Create new folder “responsegraph”. Open folder in VS Code
- In new terminal type command “gcloud init” and on prompt select google email/region/project
- Authorize access by typing “gcloud auth login”
- Screen after successful login

```
Your current project is [responsegraph]. You can change this setting by running:
$ gcloud config set project PROJECT_ID

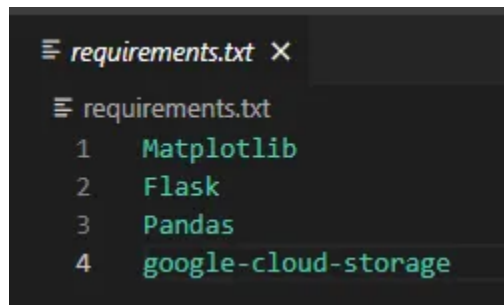
C:\workspaces\gcp\responsegraph>
```

Step 4: Lets code

- In VS Code select empty “responsegraph” folder. Below diagram shows files in folder after completing code



- Update **requirements.txt** as below



- At terminal within responsegraph folder run below commands

```
> python -m venv env
> env\Scripts\activate.bat
> pip install -r requirements.txt
```

- Update **.gcloudignore** as below

```
.gcloudignore X
.gcloudignore
1  # This file specifies files that are *not* uploaded to Google Cloud
2  # using gcloud. It follows the same syntax as .gitignore, with the addition of
3  # "#!include" directives (which insert the entries of the given .gitignore-style
4  # file at that point).
5  #
6  # For more information, run:
7  #   $ gcloud topic gcloudignore
8  #
9  .gcloudignore
10 # If you would like to upload your .git directory, .gitignore file or files
11 # from your .gitignore file, remove the corresponding line
12 # below:
13 .git
14 .gitignore
15 env
16
17 # Python pycache:
18 __pycache__/
19 # Ignored by the build system
20 /setup.cfg
```

- Update **app.yaml** as below

```
! app.yaml X
! app.yaml > ...
1  runtime: python39
2
```

- Update **util.py** as below


```

util.py > readfilefrombucket
1
2 import pandas as pd
3 from google.cloud import storage
4 from io import StringIO
5
6 def readfilefrombucket(bucketname, destinationblobname):
7
8     # connect to cloud storage
9     # download file and read into data frame
10
11     storageclient = storage.Client()
12     bucket = storageclient.bucket(bucketname)
13     blob = bucket.blob(destinationblobname)
14     blob = blob.download_as_string()
15     blob = blob.decode('utf-8')
16     blob = StringIO(blob) #transform bytes to string here
17     df = pd.read_csv(blob) #read blob into dataframe
18     return df
19
20
21 def createdf(dfAll):
22
23     # prepare dataset
24
25     df = pd.DataFrame(dfAll, columns= ['process','responsetimeorcount']) #create subset df with two columns
26     df = df.query('process.str.contains("Response Time")') #filter df to have just "response time" data
27     df['process'] = df['process'].str.slice(0, 8) # truncate the process string to display
28     dfmeanprocess = df.groupby('process').responsetimeorcount.mean().reset_index() #group and mean by unique process
29     dfmeanprocess['id'] = range(0, 0+len(dfmeanprocess)) # add new column id
30     print(dfmeanprocess)
31     return dfmeanprocess
32

```

```

33
34 def barchartvalue(ppbar, axval):
35
36     #height is value on top of bar
37
38     for p in ppbar:
39         height = p.get_height()
40         axval.annotate('{}' .format(round(height,1)),
41             xy=(p.get_x() + p.get_width() / 2, height),
42             xytext=(0, 3), # 3 points vertical offset
43             textcoords="offset points",
44             size=5.5,
45             ha='center', va='bottom',
46             )

```

- Update graph.py as below

```

graph.py > plotgraph
1  from util import barchartvalue, readfilefrombucket, createdf
2  import matplotlib.pyplot as plt
3
4
5  def plotgraph():
6      # File 1 metrics
7      file1datestr = 'Jan 4 - Jan 10 2021'
8      dfFile1 = createdf(readfilefrombucket('graphbucket_test','Jan 4 - Jan 10 2021.csv'))
9
10     # File 2 metrics
11     file2datestr = 'Jan 11 - Jan 17 2021'
12     dfFile2 = createdf(readfilefrombucket('graphbucket_test','Jan 11 - Jan 17 2021.csv'))
13
14     #File 3 metrics
15     file3datestr = 'Jan 18 - Jan 24 2021'
16     dfFile3 = createdf(readfilefrombucket('graphbucket_test','Jan 18 - Jan 24 2021.csv'))
17
18     #File 4 metrics
19     file4datestr = 'Jan 25 - Jan 31 2021'
20     dfFile4 = createdf(readfilefrombucket('graphbucket_test','Jan 25 - Jan 31 2021.csv'))
21
22
23     # plotting graph
24     fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2)
25     fig.tight_layout()
26     fig.suptitle('API transactions over 4 weeks', size=8)
27

```

```

28
29     #####File 1 Plot#####
30     ax1.set_ylabel('Seconds', size=8)
31     ax1.set_title(file1datestr, size=7)
32     ax1.set_xticks(dfFile1['id'])
33     ax1.tick_params(axis="x", labelsiz=6)
34     ax1.tick_params(axis="y", labelsiz=6)
35
36
37     ppbar1 = ax1.bar(dfFile1['id'], dfFile1['responsetimeorcount']/1000, color='royalblue', alpha=0.7)
38     ax1.grid(color='#95a5a6', linestyle='--', linewidth=2, axis='y', alpha=0.7)
39
40     barchartvalue(ppbar1,ax1) # display value on top of each bar
41

```

Explanation: (1) Call *createdf* method to load four csv files from Cloud Storage. (2) Create a 2 by 2 subplot (3) For “File 1 plot” setup axis labels and data (3) Do the same for “File 2 plot”, “File 3 plot” and “File 4 plot” as below


```

graph.py > plotgraph
43 #####File 2 Plot#####
44 ax2.set_ylabel('Seconds', size=8)
45 ax2.set_title(file2datestr, size=7)
46 ax2.set_xticks(dfFile2['id'])
47 ax2.tick_params(axis="x", labels=6)
48 ax2.tick_params(axis="y", labels=6)
49
50 ppbar2 = ax2.bar(dfFile2['id'], dfFile2['responsetimeorcount']/1000, color='royalblue', alpha=0.7)
51 ax2.grid(color='#95a5a6', linestyle='--', linewidth=2, axis='y', alpha=0.7)
52 barchartvalue(ppbar2,ax2) # display value on top of each bar
53
54
55 #####File 3 Plot#####
56 ax3.set_ylabel('Seconds', size=8)
57 ax3.set_title(file3datestr, size=7)
58 ax3.set_xticks(dfFile3['id'])
59 ax3.tick_params(axis="x", labels=6)
60 ax3.tick_params(axis="y", labels=6)
61
62 ppbar3 = ax3.bar(dfFile3['id'], dfFile3['responsetimeorcount']/1000, color='royalblue', alpha=0.7)
63 ax3.grid(color='#95a5a6', linestyle='--', linewidth=2, axis='y', alpha=0.7)
64 barchartvalue(ppbar3,ax3) # display value on top of each bar
65
66 #####File 4 Plot#####
67 ax4.set_ylabel('Seconds', size=8)
68 ax4.set_title(file4datestr, size=7)
69 ax4.set_xticks(dfFile4['id'])
70 ax4.tick_params(axis="x", labels=6)
71 ax4.tick_params(axis="y", labels=6)
72
73 ppbar4 = ax4.bar(dfFile4['id'], dfFile4['responsetimeorcount']/1000, color='royalblue', alpha=0.7)
74 ax4.grid(color='#95a5a6', linestyle='--', linewidth=2, axis='y', alpha=0.7)
75 barchartvalue(ppbar4,ax4) # display value on top of each bar

```

- In the end return “fig”

```

76
77 return fig #instead of plt.show() we return fig for creating image
78

```

- Update **main.py** — Import libraries and initialize app

```

main.py > ...
1 import io
2 from flask import Flask, Response
3 from matplotlib.backends.backend_agg import FigureCanvasAgg
4 from graph import plotgraph
5
6
7 app = Flask(__name__)
8

```

- Code **index** and **plot_png** methods

```

main.py > index
9  @app.route("/")
10 def index():
11     return f"""
12     <form method=get action="/">
13     <input type=submit value="update graph">
14     </form>
15
16     <h3>Plot as a png</h3>
17     
21     """
22 @app.route("/matplotlib-as-image.png")
23 def plot_png():
24     """ renders the plot on the fly.
25     """
26     #fig = Figure()
27     fig = plotgraph()
28
29     output = io.BytesIO()
30     FigureCanvasAgg(fig).print_png(output)
31     return Response(output.getvalue(), mimetype="image/png")
32
33 if __name__ == "__main__":
34     # Used when running locally only. When deploying to Google App
35     # Engine, a webserver process such as Gunicorn will serve the app. This
36     # can be configured by adding an `entrypoint` to app.yaml.
37     app.run(host="localhost", port=8080, debug=True)
38

```

Explanation: (1) Return html tags in *index* method. The *index* method will be called on initial page load. (2) The html *src* tag invokes *plot_png* method that generates the barchart graph by calling *plotgraph* method (in *graph.py*)

Test Local => [Continue to Part 2](#)

Matplotlib

App Engine

Written by Dev Thakkar

3 Followers · 3 Following

No responses yet



What are your thoughts?

Respond

More from Dev Thakkar

```
aml > ...  
Fact| Registry - create repository kmtestrepo  
d and push to repository kmtestrepo  
:  
e: 'gcr.io/cloud-builders/docker'  
s: [ 'build', '-t', '${_LOCATION}-docker.pkg.dev/$PROJECT_ID/${_REPOSITORY}/${_IMAGE}'  
s:  
_LOCATION}-docker.pkg.dev/$PROJECT_ID/${_REPOSITORY}/${_IMAGE}'  
stitutions:  
LOCATION: "us-east1"  
POSITORY: "kmtestrepo"  
AGE: "my-image"
```

 Dev Thakkar

Dec 29, 2021  5  1



Firestore

^

data

indexes

import/export

security rules

Data

/ > Profile

Root

+ START COLLECTION

Profile

D

 Dev Thakkar

Nov 19, 2021



You have 20 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name *

KMSTESTPROJ

Project ID *

kmstestproj



Project ID can have lowercase letters, digits, or hyphens. It must start with a lowercase letter and end with a letter or number.

D

 Dev Thakkar

Nov 19, 2021





D Dev Thakkar

Mar 15



See all from Dev Thakkar

Recommended from Medium

Always Free

24 GB RAM + 4 CPU + 200 GB



@harendraverma2



@harendra21



@harendra21



Harendra



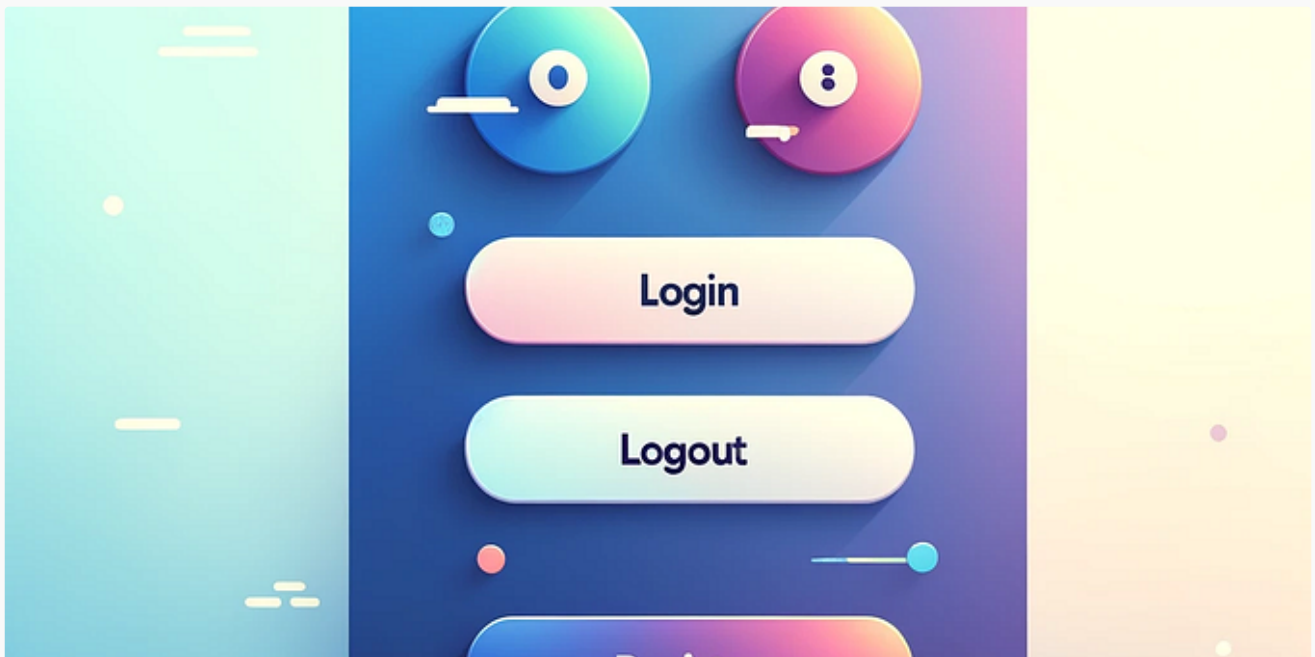
Oct 26



8.3K



132



The For Loop



3d ago



Lists



Staff picks

791 stories · 1532 saves



Stories to Help You Level-Up at Work

19 stories · 901 saves



Self-Improvement 101

20 stories · 3167 saves



Productivity 101

20 stories · 2681 saves

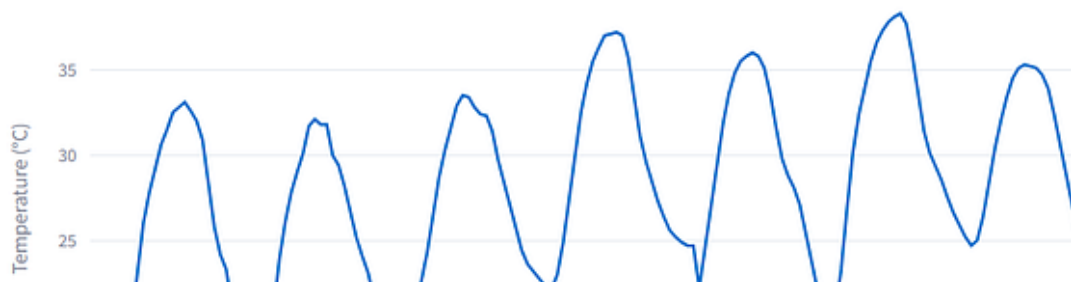
40,000000 - +

-3,000000 - +

Get Forecast

Weather forecast for Latitude: 40.0, Longitude: -3.0

Temperature Forecast



Nuno Carvalho

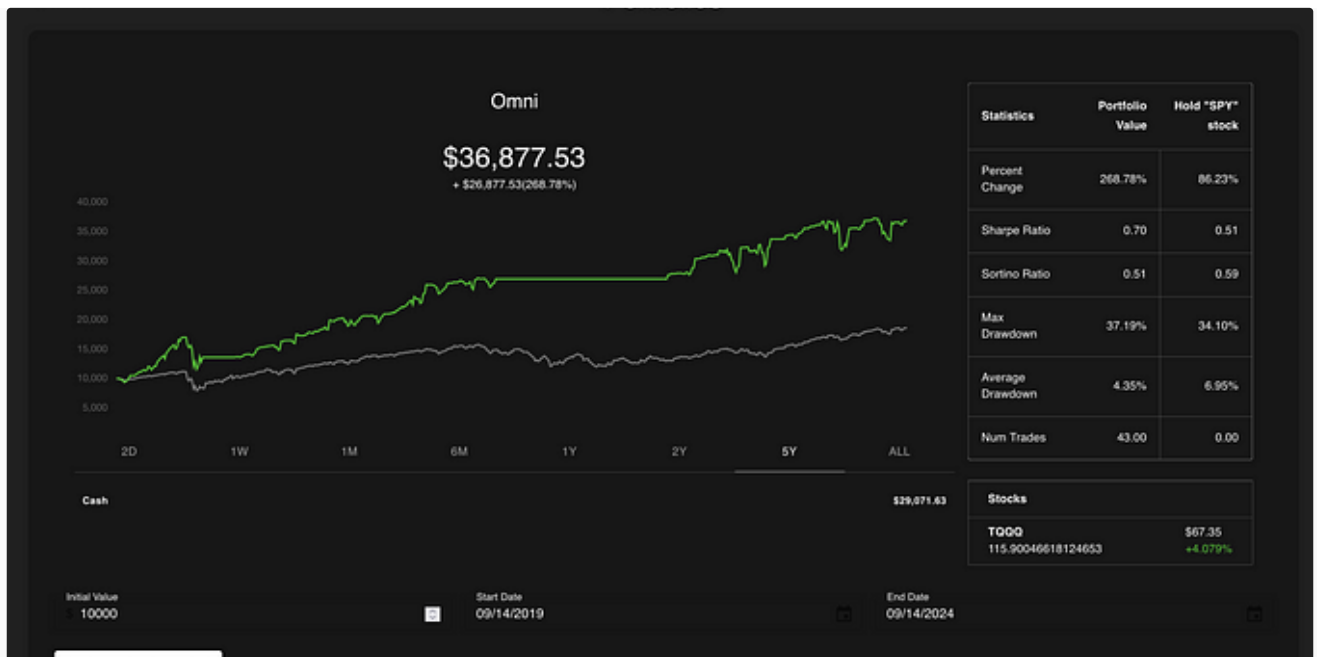



Jul 14



14






 In DataDrivenInvestor by Austin Starks

★ Sep 15 🖱️ 7.8K 💬 192



 In Towards Data Science by Sergei Savvov

3d ago 🖱️ 1.7K 💬 19





Jessica Stillman



Oct 30



18.3K



478



See more recommendations