



CAPSTONE PROJECT

Predicting Monthly Energy Needs for Trenton Falls, NY

Abstract

Using a machine learning models and ETL pipeline built with the assistance of AI, the team analyzes historical weather patterns to predict energy needs.

Jake Halsey, David Haddad, Michelle Armstrong, Appal Swamy Badi,
Michael Davis

Executive Summary:**Requirements:**

The intention is to combine at least one (in this case, 2) Archived (herein referred to as static) data sets with one live API to analyze energy consumption patterns.

Business case:

The electrical grid in the United States does not have significant energy storage capacity, and there is a cost to transporting and storing fuel for consumption. This creates a cost burden for energy companies that produce too little or too much energy in each time window.

Proposed Solution:

To facilitate the building of a future optimization model, we've built an ETL pipeline and proof-of-concept prediction model for energy consumption requirements for Trenton Falls, NY. By Analyzing historical weather and energy consumption data, we attempt to use live weather data for zip codes in Trenton Falls, NY, to estimate the likely energy needs in the current month. Using a mature prediction model to create an optimization model, energy producers could reduce storage and transportation costs.

Key Findings:

We find a strong, negative correlation (-0.82) between temperature and energy consumption, suggesting that colder months require significantly more energy than cooler months.

Model Strengths:

While temperature is not the only factor in determining usage, many other prediction models, such as predictions based on cell phone usage, cannot be adjusted by seasonal factors, affecting their reliability. We believe our model could be used to augment such models in the future, addressing the limitations of both models.

Model Limitations:

Few states track energy data below the state level, and only a limited number of weather stations provide local temperature records. Our model can predict energy needs for areas with both data points but not for those without. Correlations will likely vary by location, requiring the model to adjust accordingly. For example, while the current model suits New York's climate, a different pattern might emerge in warmer places like Florida. Further research is needed to refine the model.

Recommendations:

To enhance consumption prediction, local weather and energy data should be closely monitored and integrated to expand the model to more locations. Where data exists, the model should augment current systems for greater accuracy. The final model should then be used to create an optimization model for utility production.

Appendix:

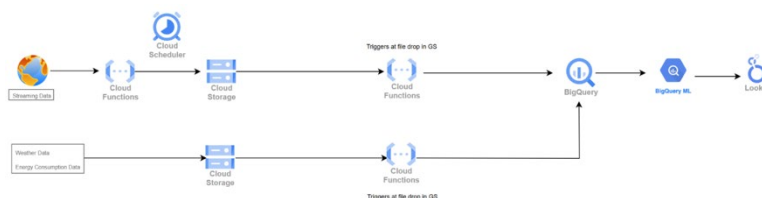
Project Resources

- **Static Data:**
 - Historical Energy Data: [https://statics.teams.cdn.office.net/evergreen-assets/safelinks/1/atp-safelinks.html\]\(https://catalog.data.gov/dataset/utility-energy-registry-monthly-zip-code-energy-use-beginning-2016](https://statics.teams.cdn.office.net/evergreen-assets/safelinks/1/atp-safelinks.html](https://catalog.data.gov/dataset/utility-energy-registry-monthly-zip-code-energy-use-beginning-2016)
 - Historical Weather Data: <https://www.climate.gov/maps-data/dataset/past-weather-zip-code-data-table>
- **Live Data**
 - Weather
API: <https://api.weatherapi.com/v1/current.json?key=7c8218f18550417496b43123242902&q={area}>
- **Competing Models (Cell phone data)**
 - <https://epjdatascience.springeropen.com/articles/10.1140/epjds/s13688-016-0075-3#:~:text=An%20accurate%20prediction%20of%20energy,allowing%20an%20efficient%20energy%20storage.>
- **Github Repository Link:** https://github.com/jhalsey87/BigData_Capstone824/tree/main
- **Lookerstudio Dashboard:** <https://lookerstudio.google.com/u/0/reporting/00b216a9-b9e3-4ba4-a8a1-739f389aefa4/page/KAd8D>

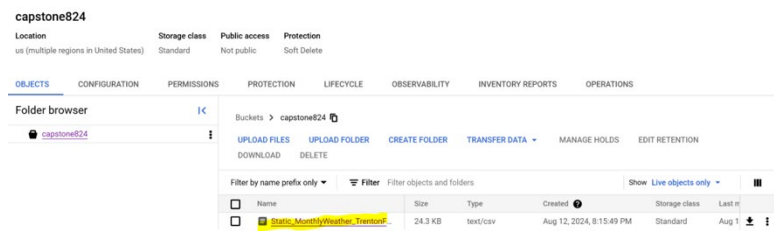
Methodologies

- **Data Integration:** Combined historical weather and energy consumption data with real-time weather information via a live API.
- **Correlation Analysis:** The relationship between temperature and energy consumption was assessed to identify actionable patterns.
- **Predictive Modeling:** Developed a proof-of-concept model to project energy consumption based on live weather inputs, serving as a foundation for enhancing future prediction accuracy.

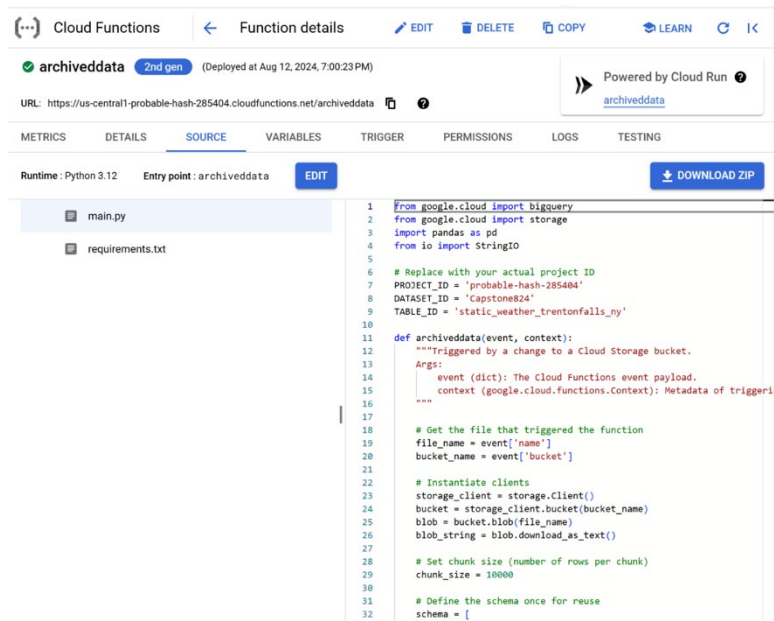
Pipeline:



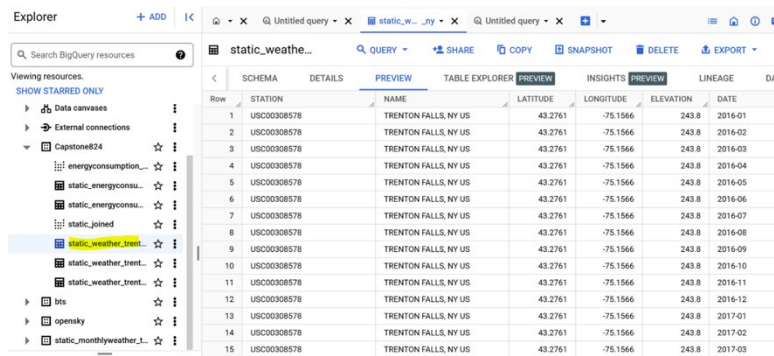
Static Monthly Weather Archival Data Loaded Into Cloud Storage:



Cloud Function for a One-Time Load of Data from Cloud Storage to Big Query Table:



Data Loaded Into Big Query Table:



Log Extract of Cloud Function Trigger from Cloud Storage File Upload:

Cloud Functions

Function details

EDIT DELETE COPY LEARN

archiveddata 2nd gen (Deployed at Aug 12, 2024, 7:00:23 PM)

Powered by Cloud Run

URL: https://us-central1-probable-hash-285404.cloudfunctions.net/archiveddata

METRICS DETAILS SOURCE VARIABLES TRIGGER PERMISSIONS LOGS TESTING

Logs Severity: Default Filter Search all fields and values

SEVERITY	TIMESTAMP	SUMMARY
>	2024-08-12 18:51:36.760 PDT	Error converting Pandas column with name: 'COSO_ATTRIBUTES' and datatype: 'int64' to an appropriat...
>	2024-08-12 18:51:36.777 PDT	Exception on / [POST] Traceback (most recent call last): File "/layers/google.python.pip/pip/lib...
>	2024-08-12 18:59:34.681 PDT	namespaces/probable-hash-285404/servi... service-7574969733538gcf-admin-robot... (@type: type...
>	2024-08-12 18:59:35.232 PDT	Cloud Functions UpdateFunction us-central1:archiveddata appalbad@gmail.com (@type: type.googl...
>	2024-08-12 19:00:13.966 PDT	namespaces/probable-hash-285404/servi... service-7574969733538gcf-admin-robot... (@type: type...
>	2024-08-12 19:00:22.032 PDT	Default STARTUP TCP probe succeeded after 1 attempt for container "worker" on port 8080.
>	2024-08-12 19:00:22.107 PDT	Cloud Run v1 archiveddata-00017-zuh (@type: type.googleapis.com/google.cloud.audit.AuditLog, ...
>	2024-08-12 19:00:23.501 PDT	Cloud Run v1 archiveddata (@type: type.googleapis.com/google.cloud.audit.AuditLog, methodName...
>	2024-08-12 19:00:23.846 PDT	Cloud Functions UpdateFunction us-central1:archiveddata appalbad@gmail.com (@type: type.googl...
>	2024-08-12 19:00:41.760 PDT	Default STARTUP TCP probe succeeded after 1 attempt for container "worker" on port 8080.
>	2024-08-12 19:00:41.806 PDT	POST 200 130 B 3 s APIs-Google; (+https://developers.goo... https://archiveddata-vdukgnrzaq-uc...
>	2024-08-12 19:00:44.848 PDT	Successfully loaded data into Capstone824:static_weather_trentonfalls.ny from file Static_MonthlyW...
>	2024-08-12 19:04:55.872 PDT	POST 200 331 B 683 ms APIs-Google; (+https://developers.goo... https://archiveddata-vdukgnrzaq...
>	2024-08-12 19:04:56.572 PDT	Exception on / [POST] Traceback (most recent call last): File "/layers/google.python.pip/pip/lib...
>	2024-08-12 20:15:54.510 PDT	Default STARTUP TCP probe succeeded after 1 attempt for container "worker" on port 8080.
>	2024-08-12 20:15:58.388 PDT	Successfully loaded data into Capstone824:static_weather_trentonfalls.ny from file Static_MonthlyW...

Energy Archival Data Loaded Into Cloud Storage:

capstone824

Location: us (multiple regions in United States) Storage class: Standard Public access: Not public Protection: Soft Delete

OBJECTS CONFIGURATION PERMISSIONS PROTECTION LIFECYCLE OBSERVABILITY INVENTORY REPORTS OPERATIONS

Folder browser

capstone824

Upload FILES Upload Folder Create Folder Transfer Data Manage Holds Edit Retention

Filter by name prefix only Filter Filter objects and folders Show Live objects only

Name	Size	Type	Created	Storage class	Last mo
Static_MonthlyWeather_TrentonF...	24.3 KB	text/csv	Aug 12, 2024, 8:15:40 PM	Standard	Aug 12, ...
Static_Energy_Consumption_Monthly...	1.6 KB	text/csv	Aug 12, 2024, 9:01:19 PM	Standard	Aug 12, ...
weather_data_schema.json	5.9 KB	application/json	Aug 12, 2024, 12:27:55 PM	Standard	Aug 12, ...

Cloud Function For A One Time Load Of Data From Cloud Storage To Big Query Table:

weatherdata 2nd gen (Deployed at Aug 12, 2024, 9:06:31 PM)

Powered by Cloud Run

URL: https://us-central1-probable-hash-285404.cloudfunctions.net/weatherdata

METRICS DETAILS SOURCE VARIABLES TRIGGER PERMISSIONS LOGS TESTING

Runtime: Python 3.12 Entry point: weatherdata EDIT DOWNLOAD ZIP

main.py requirements.txt

```
1 from google.cloud import bigquery
2 from google.cloud import storage
3 import pandas as pd
4 from io import StringIO
5
6 # Replace with your actual project ID
7 PROJECT_ID = 'probable-hash-285404'
8 DATASET_ID = 'Capstone824'
9 TABLE_ID = 'static_energyconsumptionbyzip_NY'
10
11 def weatherdata(event, context):
12     """Triggered by a change to a Cloud Storage bucket.
13     Args:
14         event (dict): The Cloud Functions event payload.
15         context (google.cloud.functions.Context): Metadata of trigger
16
17
18     # Get the file that triggered the function
19     file_name = event['name']
20     bucket_name = event['bucket']
21
22     # Instantiate clients
23     storage_client = storage.Client()
24     bucket = storage_client.bucket(bucket_name)
25     blob = bucket.blob(file_name)
26     blob_string = blob.download_as_text()
27
28     # Set chunk size (number of rows per chunk)
29     chunk_size = 10000
30
31     # Define the schema once for reuse
32     schema = [
```

Cloud Function Trigger Log:

weatherdata

2nd gen

(Deployed on Aug 12, 2024, 9:06:31 PM)

Powered by Cloud Run

URL: https://us-central1-probable-hash-285404.cloudfunctions.net/weatherdata

METRICS

DETAILS

SOURCE

VARIABLES

TRIGGER

PERMISSIONS

LOGS

TESTING

Logs

Severity: Default

Filter: Search all fields and values

SEVERITY	TIMESTAMP	SUMMARY
> i	2024-08-12 21:00:59.605 PDT	Cloud Run v1 weatherdata-00004-nec (@type: type.googleapis.com/google.cloud.audit.AuditLog, m...
> i	2024-08-12 21:01:01.004 PDT	Cloud Run v1 weatherdata (@type: type.googleapis.com/google.cloud.audit.AuditLog, methodName:...
> i	2024-08-12 21:01:01.505 PDT	Cloud Functions UpdateFunction us-central1:weatherdata appalbadi@gmail.com (@type: type.google...
> i	2024-08-12 21:01:20.683 PDT	POST 200 331 B 288 ms APIs-Google; (<https://developers.goo... https://weatherdata-vdugnrzaq-...
> *	2024-08-12 21:01:20.988 PDT	Exception on / [POST] Traceback (most recent call last): File "/layers/google.python.pip/pip/lib...
> i	2024-08-12 21:05:38.088 PDT	namespaces/probable-hash-285404/servi... service-757496973353gcf-admin-robot... (@type: type...
> i	2024-08-12 21:05:38.730 PDT	Cloud Functions UpdateFunction us-central1:weatherdata appalbadi@gmail.com (@type: type.google...
> i	2024-08-12 21:06:20.024 PDT	namespaces/probable-hash-285404/servi... service-757496973353gcf-admin-robot... (@type: type...
> i	2024-08-12 21:06:29.196 PDT	Default STARTUP TCP probe succeeded after 1 attempt for container "worker" on port 8080.
> i	2024-08-12 21:06:29.269 PDT	Cloud Run v1 weatherdata-00005-yit (@type: type.googleapis.com/google.cloud.audit.AuditLog, m...
> i	2024-08-12 21:06:30.658 PDT	Cloud Run v1 weatherdata (@type: type.googleapis.com/google.cloud.audit.AuditLog, methodName:...
> i	2024-08-12 21:06:31.077 PDT	Cloud Functions UpdateFunction us-central1:weatherdata appalbadi@gmail.com (@type: type.google...
> i	2024-08-12 21:06:59.011 PDT	POST 200 130 B 3.9 s APIs-Google; (<https://developers.goo... https://weatherdata-vdugnrzaq-u...
> *	2024-08-12 21:07:02.945 PDT	Successfully loaded data into Capstone624:static_energyconsumptionbyzip_NY from file Utility_Energ...

Big Query Table Loaded Through Cloud Function Trigger:

Explorer

+ ADD

IK

static_en_..._NY

static_en_...

Untitled query

Untitled query

static_en_...

QUERY

SHARE

COPY

SNAPSHOT

DELETE

EXPORT

R

SCHEMA

DETAILS

PREVIEW

TABLE EXPLORER

INSIGHTS

LINEAGE

DATA PROFIL

Row	year	data_class	data_field_display_name	data_field	zip_city
1	2020	electricity	Business Consumption (SC+G)	2_nat_consumption	Congo Lake
2	2020	electricity	Business Consumption (SC+G)	2_nat_consumption	Congo Lake
3	2020	electricity	ICAP Capacity Tag (R)	1_my_policy_support	Congo Lake
4	2020	electricity	CCA-Ineligible Customer Count...	8_my_policy_support	Congo Lake
5	2020	electricity	ICAP Capacity Tag (R)	1_my_policy_support	Congo Lake
6	2020	electricity	ICAP Capacity Tag (T)	6_my_policy_support	Congo Lake
7	2020	electricity	ICAP Capacity Tag (T)	6_my_policy_support	Congo Lake
8	2020	electricity	ICAP Capacity Tag (SC+G)	5_my_policy_support	Congo Lake
9	2020	electricity	CCA-Ineligible Customer Count...	8_my_policy_support	Congo Lake
10	2020	electricity	ICAP Capacity Tag (T)	6_my_policy_support	Glenfield
11	2020	electricity	ICAP Capacity Tag (R)	1_my_policy_support	Glenfield
12	2020	electricity	CCA-Ineligible Customer Count...	7_my_policy_support	Pittard
13	2020	electricity	ICAP Capacity Tag (SC)	2_my_policy_support	Pittard
14	2020	electricity	CCA-Ineligible Customer Count...	8_my_policy_support	Pittard
15	2020	electricity	CCA-Ineligible Customer Count...	8_my_policy_support	Pittard
16	2020	electricity	Total Consumption (T)	3_nat_consumption	Pittard
17	2020	natural_gas	Business Consumption (SC+G)	5_nat_consumption	Castleton on Hudson
18	2020	electricity	Total Consumption (T)	3_nat_consumption	Castleton on Hudson
19	2020	electricity	ICAP Capacity Tag (R)	1_my_policy_support	Castleton on Hudson
20	2020	electricity	CCA-Ineligible Customer Count...	7_my_policy_support	Castleton on Hudson

static_energyconsumptionbyzip_NY

probable-hash-285404.Capstone624

Last modified: Aug 12, 2024, 9:10:19 PM UTC-7

Streaming Data Load Cloud Function to Call Apis and Load The Data Into Cloud Storage:

Cloud Functions

Function details

EDITDELETECOPY

streamdata2nd gen

(Deployed at Aug 13, 2024, 11:34:08 AM)

URL: https://us-central1-probable-hash-285404.cloudfunctions.net/streamdata

METRICS

DETAILS

SOURCE

VARIABLES

TRIGGER

PERMISSIONS

LOGS

TESTING

Runtime: Python 3.12

Entry point: streamdata

EDIT

main.py

requirements.txt

```
1 import functions_framework
2 import requests
3 import pandas as pd
4 from google.cloud import storage
5 from datetime import datetime
6 import os
7
8 # GCP Cloud Function entry point
9
10 def streamdata(request):
11     # Extract api_keys and areas from the request (query string parameters)
12     api_keys = ['7c8218f18550417496b43123242982','7c8218f18550417496b43123242982','7c8218f1
13     areas = ['13354','13304','13352','13435','13438','13469','13438']
14
15     if not api_keys or not areas:
16         return "api_key and area parameters are required", 400
17
18     if len(api_keys) != len(areas):
19         return "The number of api_key and area values must be the same", 400
20
21     # Initialize Cloud Storage client
22     storage_client = storage.Client()
23     bucket_name = "capstone_streaming" # Replace with your bucket name
24     bucket = storage_client.bucket(bucket_name)
25
26     for api_key, area in zip(api_keys, areas):
27         # WeatherAPI URL
28         url = f"https://api.weatherapi.com/v1/current.json?key={api_key}&q={area}"
29
30         # Make the API request
31         response = requests.get(url)
32         if response.status_code != 200:
```

Cloud Function to Load Data from Cloud Storage to Bigquery On The Cloud Storage Trigger:

Cloud Functions

Function details

EDITDELETECOPY

streamdataload2nd gen

(Deployed at Aug 13, 2024, 12:28:41 PM)

URL: https://us-central1-probable-hash-285404.cloudfunctions.net/streamdataload

METRICS

DETAILS

SOURCE

VARIABLES

TRIGGER

PERMISSIONS

LOGS

TESTING

Runtime: Python 3.12

Entry point: streamdataload

EDIT

main.py

requirements.txt

```
1 from google.cloud import bigquery
2 from google.cloud import storage
3 import pandas as pd
4 from io import StringIO
5
6 # Replace with your actual project ID
7 PROJECT_ID = 'probable-hash-285404'
8 DATASET_ID = 'capstone@24'
9 TABLE_ID = 'streaming_data'
10
11 def streamdataload(event, context):
12     """Triggered by a change to a Cloud Storage bucket.
13     Args:
14         event (dict): The Cloud Functions event payload.
15         context (google.cloud.functions.Context): Metadata of triggering event.
16     """
17
18     # Get the file that triggered the function
19     file_name = event['name']
20     bucket_name = event['bucket']
21
22     # Instantiate clients
23     storage_client = storage.Client()
24     bucket = storage_client.bucket(bucket_name)
25     blob = bucket.blob(file_name)
26     blob_string = blob.download_as_text()
27
28     # Set chunk size (number of rows per chunk)
29     chunk_size = 100000
30
31     # Define the schema once for reuse
32     schema = [
```

Scheduler to Run the API Call Function Every 5 Minutes:

Cloud Scheduler

Jobs

CREATE JOB

REFRESH

FORCE RUN

EDIT

COPY

PAUSE

RESUME

DELETE

LEARN

SCHEDULER JOBS

APP ENGINE CRON JOBS

Filter

Filter jobs

Name	Status of last execution	Region	State	Description	Frequency	Target	Last run	Next run	Last updated
opendata	Has not run yet	us-central1	Paused	opendata	5 * * * * (America/Los_Angeles)	URL: https://us-central1-probable-hash-285404.cloudfunctions.net/opendata	Aug 5, 2024, 5:17:10 PM	Aug 5, 2024, 5:53:50 PM	Aug 5, 2024, 5:53:50 PM
streamdata	Success	us-central1	Enabled	streamdata	5 * * * * * (America/Los_Angeles)	URL: https://us-central1-probable-hash-285404.cloudfunctions.net/streamdata	Aug 13, 2024, 12:36:19 PM	Aug 13, 2024, 1:05:00 PM	Aug 13, 2024, 12:34:08 PM

Big Query Table Loading Streaming Data

streaming_data

Row	name	region	country	lat	lon	tz_id
1	Holland Patent	New York	USA	43.23	-75.27	America/New_York
2	Rensselaer	New York	USA	43.33	-75.2	America/New_York
3	Prospect	New York	USA	43.3	-75.15	America/New_York
4	Holland Patent	New York	USA	43.23	-75.27	America/New_York

Looker Dashboard (2 Pages)

