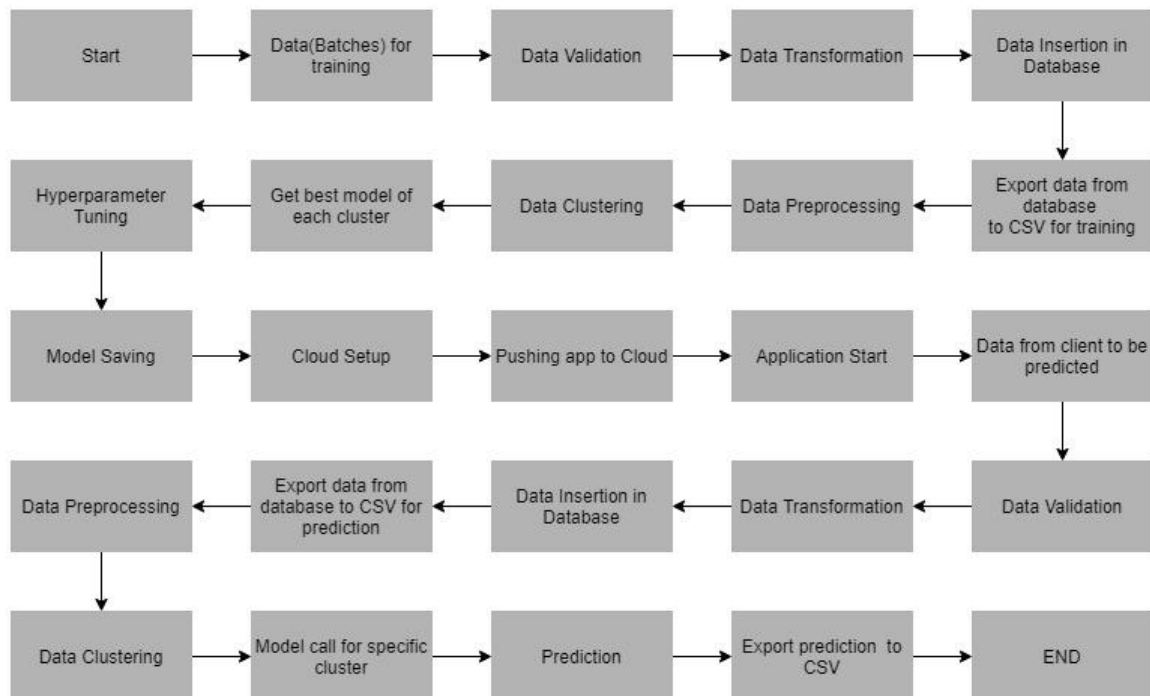


Problem Statement

To build a regression model to predict the heating load and cooling load based on the different features in the training data.

Architecture



Data Description

Given is the variable name, variable type, the measurement unit and a brief description.

The “EcoComfort: Intelligent Energy Efficiency” is the regression problem. The order of this listing corresponds to the order of numerals along the rows of the database.

Name	Data Type	Measurement	Description
Relative Compactness	quantitative	Fraction Between (0-1)	Input Variable - Relative compactness is a dimensionless quantity and does not have a specific unit. It

			is a ratio or a coefficient comparing the actual surface area of the building to the surface area of a hypothetical cuboid with the same volume.
Surface Area	quantitative	square meters (m ²)	Input Variable - It represents the total area of the building envelope, including walls, roof, windows, and other external surfaces.
Wall Area	quantitative	square meters (m ²)	Input Variable- refers to the total surface area of the walls of the building.
Roof Area	quantitative	square meters (m ²)	Input Variable – It represents the total surface area of the roof of the building.
Overall Height	quantitative	meters (m)	Input Variable-- It represents the vertical distance from the base to the highest point of the building.
Orientation	quantitative	Number	Input Variable-- It refers to the direction in which the building is facing or positioned, such as north, south, east, or west.

Glazing Area	quantitative	square meters (m ²)	Input Variable— It represents the total area of windows and other transparent or translucent surfaces in the building envelope.
Glazing Area Distribution	quantitative	percentage or a fraction	Input Variable – It is a measure of how the glazing area is distributed or located on different facades of the building.
Heating Load (HL)	quantitative	kilowatts (kW)	Output Variable - The heating load is a measure of the amount of heat energy required to maintain a comfortable indoor temperature during the heating season
Cooling Load (CL)	Quantitative	Kilowatts (kW)	Output Variable - The cooling load is a measure of the amount of cooling capacity required to maintain a comfortable indoor temperature during the cooling season

Apart from training files, we also require a "schema" file from the client, which contains all the relevant information about the training files such as:

Name of the files, Length of Date value in FileName, Length of Time value in FileName, Number of Columns, Name of the Columns, and their datatype.

Data Validation

In this step, we perform different sets of validation on the given set of training files.

1. Name Validation- We validate the name of the files based on the given name in the schema file. We have created a regex pattern as per the name given in the schema file to use for validation. After validating the pattern in the name, we check for the length of date in the file name as well as the length of time in the file name. If all the values are as per requirement, we move such files to “Good_Raw” folder else we move such files to “Bad_Raw” folder.
2. Number of Columns - We validate the number of columns present in the files, and if it doesn't match with the value given in the schema file, then the file is moved to “Bad_Raw” folder.
3. Name of Columns - The name of the columns is validated and should be the same as given in the schema file. If not, then the file is moved to “Bad_Raw” folder.
4. The datatype of columns - The datatype of columns is given in the schema file. This is validated when we insert the files into Database. If the datatype is wrong, then the file is moved to “Bad_Raw” folder.
5. Null values in columns - If any of the columns in a file have all the values as NULL or missing, we discard such a file and move it to “Bad_Raw” folder.

Data Insertion in Database

- 1) Database Creation and connection - Create a database with the given name passed. If the database is already created, open the connection to the database.
- 2) Table creation in the database - Table with name - "Good_Raw_Data", is created in the database for inserting the files in the “Good_Raw” folder based on given column names and datatype in the schema file. If the table is already present, then the new table is not created and new files are inserted in the already present table as we want training to be done on new as well as old training files.

3) Insertion of files in the table - All the files in the “Good_Raw” folder are inserted in the above-created table. If any file has invalid data type in any of the columns, the file is not loaded in the table and is moved to “Bad_Raw” folder.

Model Training

1) Data Export from Db - The data in a stored database is exported as a CSV file to be used for model training.

2) Data Preprocessing

a) Check for null values in the columns. If present, impute the null values using the KNN imputer

b) transform the features using log transformation.

c) Scale the training and test data separately.

d) Perform Principal Component analysis and convert the features into principal components.

3) Clustering - KMeans algorithm is used to create clusters in the preprocessed data. The optimum number of clusters is selected by plotting the elbow plot, and for the dynamic selection of the number of clusters, we are using "KneeLocator" class. The idea behind clustering is to implement different algorithms to train data in different clusters. The Kmeans model is trained over preprocessed data and the model is saved for further use in prediction.

4) Model Selection - After clusters are created, we find the best model for each cluster. We are using two algorithms, "Random forest Regressor" and “Linear Regression”. For each cluster, both the algorithms are passed with the best parameters derived from GridSearch. We calculate the Rsquared scores for both models and select the model with the best score. Similarly, the model is selected for each cluster. All the models for every cluster are saved for use in prediction.

Prediction Data Description

Client will send the data in multiple set of files in batches at a given location. Data will contain 8 columns.

Apart from prediction files, we also require a "schema" file from client which contains all the relevant information about the training files such as:

Name of the files, Length of Date value in FileName, Length of Time value in FileName, Number of Columns, Name of the Columns and their datatype.

Data Validation

In this step, we perform different sets of validation on the given set of prediction files.

- 1) Name Validation- We validate the name of the files on the basis of given Name in the schema file. We have created a regex pattern as per the name given in schema file, to use for validation. After validating the pattern in the name, we check for length of date in the file name as well as length of time in the file name. If all the values are as per requirement, we move such files to “Good_Raw” folder else we move such files to “Bad_Raw” folder.
- 2) Number of Columns - We validate the number of columns present in the files, if it doesn't match with the value given in the schema file then the file is moved to “Bad_Raw” folder.
- 3) Name of Columns - The name of the columns is validated and should be same as given in the schema file. If not, then the file is moved to “Bad_Raw” folder.
- 4) Datatype of columns - The datatype of columns is given in the schema file. This is validated when we insert the files into Database. If datatype is wrong then the file is moved to “Bad_Raw” folder.
- 5) Null values in columns - If any of the columns in a file has all the values as NULL or missing, we discard such file and move it to “Bad_Raw” folder.

Data Insertion in Database

- 1) Database Creation and connection - Create database with the given name passed. If the database is already created, open the connection to the database.
- 2) Table creation in the database - Table with name - "Good_Data", is created in the database for inserting the files in the “Good_Raw” folder on the basis of given column names and datatype in the schema file. If table is already present then new table is not created, and new files are inserted the already present table as we want training to be done on new as well old training files.
- 3) Insertion of files in the table - All the files in the “Good_Raw” folder are inserted in the above-created table. If any file has invalid data type in any of the columns, the file is not loaded in the table and is moved to “Bad_Raw” folder.

Prediction

- 1) Data Export from Db - The data in the stored database is exported as a CSV file to be used for prediction.
- 2) Data Preprocessing
 - a) Check for null values in the columns. If present, impute the null values using the KNN imputer
 - b) transform the features using log transformation
 - c) Scale the training and test data separately
 - d) Perform Principal Component analysis and convert the features into principal components.
- 3) Clustering - KMeans model created during training is loaded, and clusters for the preprocessed prediction data is predicted.
- 4) Prediction - Based on the cluster number, the respective model is loaded and is used to predict the data for that cluster.
- 5) Once the prediction is made for all the clusters, the predictions along with the original names before label encoder are saved in a CSV file at a given location and the location is returned to the client.

Deployment

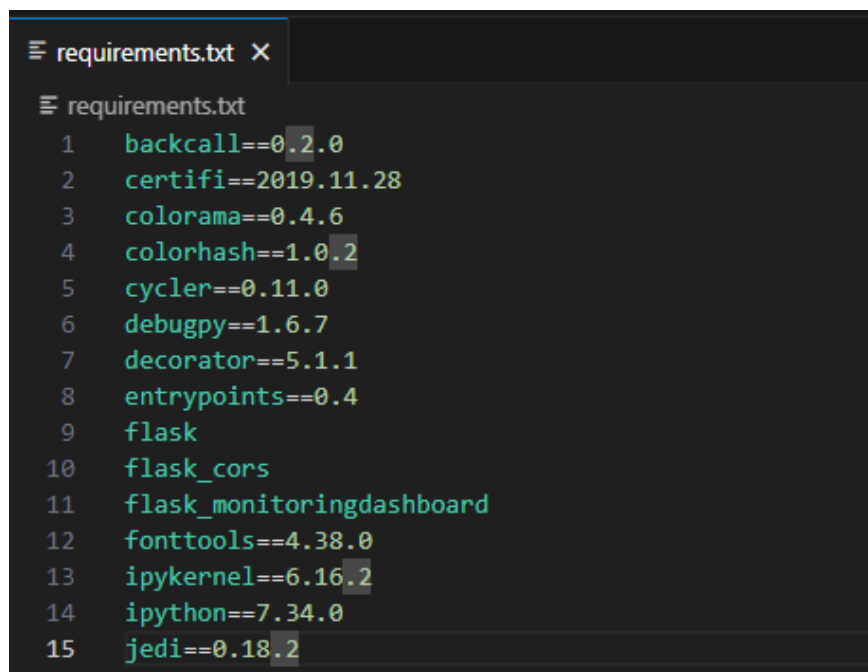
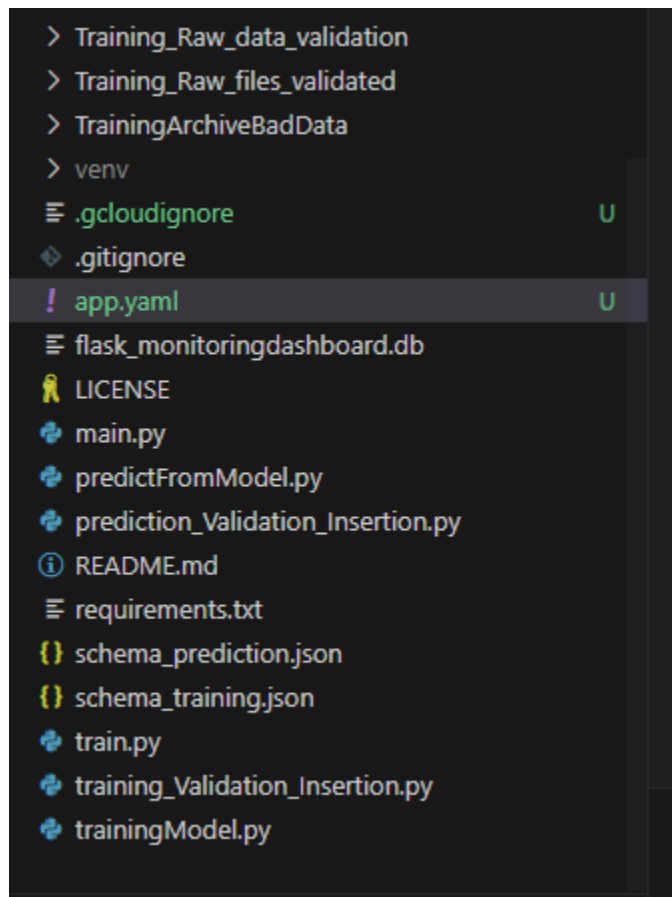
We will be deploying the model to the Google Cloud Platform.

This is a workflow diagram for the prediction of using the trained model.

Now let's see the "EcoComfort: Intelligent Energy Efficiency" project folder structure.

✓ ECOCOMFORT-INTELLIGENT-ENERGY-EFFICIENCY-FOR-...

- > __pycache__
- > application_logging
- > best_model_finder
- > data_ingestion
- > data_preprocessing
- > DataTransform_Training
- > DataTransformation_Prediction
- > DataTypeValidation_Insertion_Prediction
- > DataTypeValidation_Insertion_Training
- > EDA
- > file_operations
- > models
- > Prediction_Batch_files
- > Prediction_Database
- > Prediction_FileFromDB
- > Prediction_Logs
- > Prediction_Output_File
- > Prediction_Raw_Data_Validation
- > Prediction_Raw_Files_Validated
- > PredictionArchivedBadData
- > preprocessing_data
- > templates
- > Training_Batch_Files
- > Training_Database
- > Training_FileFromDB
- > Training_Logs



requirements.txt file consists of all the packages that you need to deploy the app in the cloud.

```

main.py > ...
25 @app.route("/predict", methods=['POST'])
26 @cross_origin()
27 def predictRouteClient():
28     try:
29         if request.json is not None:
30             path = request.json['filepath']
31
32             pred_val = pred_validation(path) #object initialization
33
34             pred_val.prediction_validation() #calling the prediction_validation function
35
36             pred = prediction(path) #object initialization
37
38             # predicting for dataset present in database
39             path = pred.predictionFromModel()
40             return Response("Prediction File created at %s!!!" % path)
41         elif request.form is not None:
42             path = request.form['filepath']
43
44             pred_val = pred_validation(path) #object initialization
45
46             pred_val.prediction_validation() #calling the prediction_validation function
47
48             pred = prediction(path) #object initialization
49
50             # predicting for dataset present in database
51             path = pred.predictionFromModel()
52             return Response("Prediction File created at %s!!!" % path)
53
54     except ValueError:
55         return Response("Error Occurred! %s" %ValueError)
56     except KeyError:

```

main.py is the entry point of our application, where the flask server starts.

```
predictFromModel.py X
predictFromModel.py > ...
1  import pandas
2  from file_operations import file_methods
3  from data_preprocessing import preprocessing
4  from data_ingestion import data_loader_prediction
5  from application_logging import logger
6  from Prediction_Raw_Data_Validation.predictionDataValidation import Prediction_Data_validation
7
8
9
10 class prediction:
11
12     def __init__(self,path):
13         self.file_object = open("Prediction_Logs/Prediction_Log.txt", 'a+')
14         self.log_writer = logger.App_Logger()
15         self.pred_data_val = Prediction_Data_validation(path)
16
17     def predictionFromModel(self):
18
19         try:
20             self.pred_data_val.deletePredictionFile() #deletes the existing prediction file from last run!
21             self.log_writer.log(self.file_object,'Start of Prediction')
22             data_getter=data_loader_prediction.Data_Getter_Pred(self.file_object,self.log_writer)
23             data=data_getter.get_data()
24
25
26             preprocessor=preprocessing.Preprocessor(self.file_object,self.log_writer)
27
28             is_null_present,cols_with_missing_values=preprocessor.is_null_present(data)
29             if(is_null_present):
30                 data=preprocessor.impute_missing_values(data)
31
32             data = preprocessor.logTransformation(data)
33
34         except Exception as e:
35             self.log_writer.log(self.file_object,e)
```

This is the **predictFromModel.py** file where the predictions take place based on the data we are giving as an input to the model.

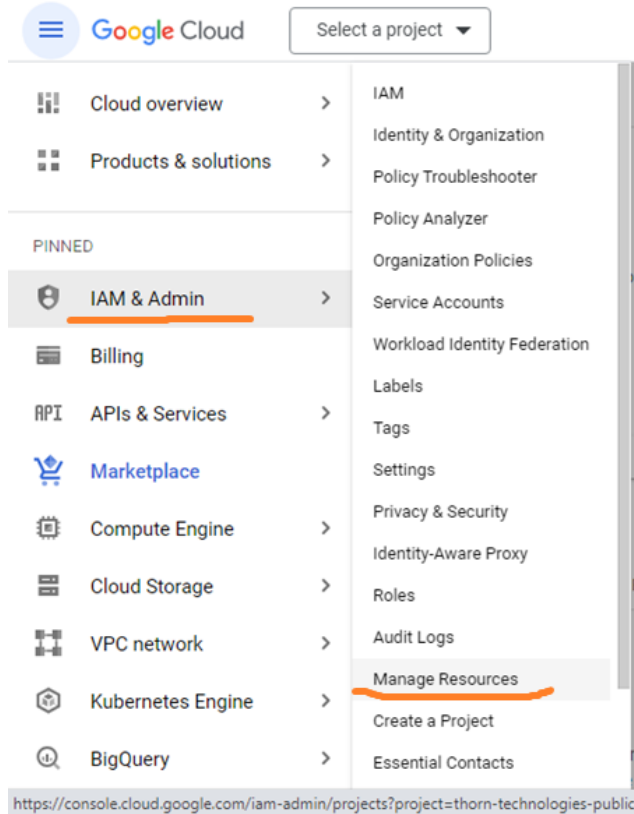
```
! app.yaml X
! app.yaml
1  runtime: python37
```

app.yaml:- It contains the Python version number.

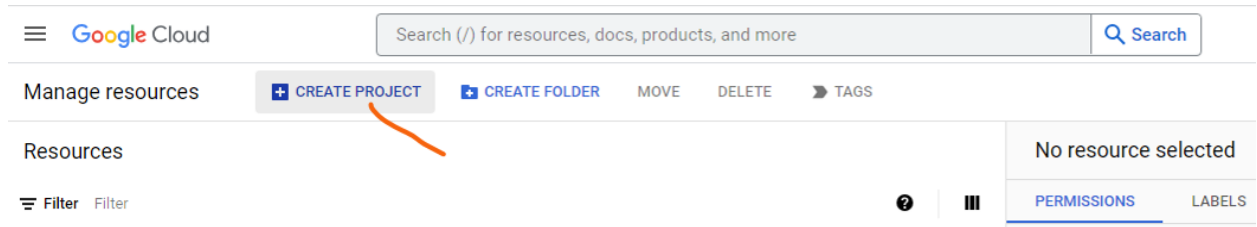
Deployment to Google cloud (GCP)

1. Go to <https://cloud.google.com/> and create an account if you haven't created one. Then go to the console of your account. <https://console.cloud.google.com/>

2. Go to IAM & Admin and select Manage Resources



3. In the manage resources page, click on CREATE PROJECT option.



4. Give your project name and Location details. And click on CREATE.

Google Cloud

Search (/) for resources, docs, products, and more

Search

New Project

You have 19 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name *
energy-efficiency

Project ID: rock-atlas-393006. It cannot be changed later. [EDIT](#)

Location *
No organization [BROWSE](#)

Parent organization or folder

CREATE CANCEL

5. Type app engine in search box and open it.

Google Cloud

app engine

Search

Manage resources [+ CREATE PROJECT](#)

Resources

Filter Filter

Filter

NAME ID

No organization

PRODUCTS & PAGES

- Application Settings
App Engine
- App Engine
Managed app platform
- Compute Engine
VMs, GPUs, TPUs, disks

Google Cloud

EcoComfort-energy-efficiency

app engine

Search

App Engine

Dashboard

Services

Versions

Instances

Task queues

Cron jobs

Security scans

Firewall rules

Quotas

Release Notes

Welcome to App Engine

Build scalable apps in any language on Google's infrastructure

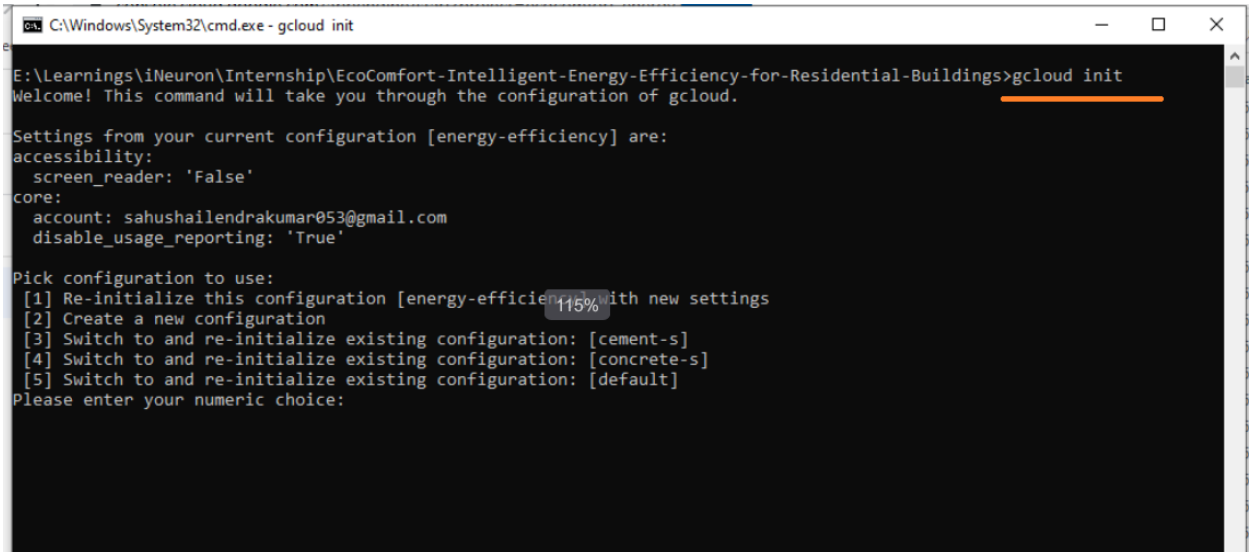
CREATE APPLICATION

Now viewing project "EcoComfort-energy-efficiency" in organization "No organization"

[More Options](#)

6. Go to <https://dl.google.com/dl/cloudsdk/channels/rapid/GoogleCloudSDKInstaller.exe> to download the google cloud SDK in your machine. And install it.

7. Go to the local folder of the project in your machine and launch cmd in that folder.
8. Enter the command gcloud init to initialise the gcloud context.



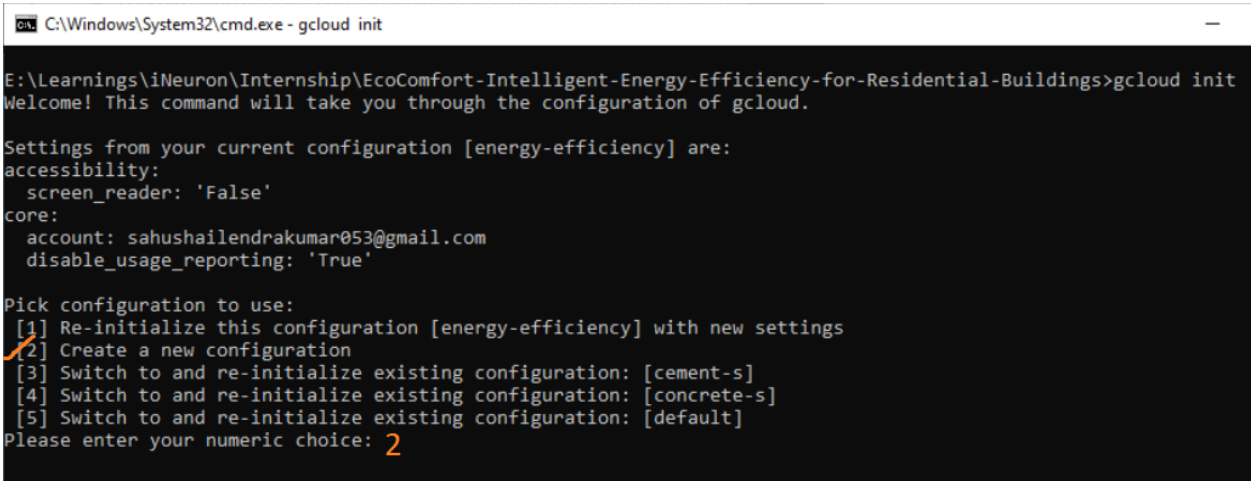
```
C:\Windows\System32\cmd.exe - gcloud init

E:\Learnings\iNeuron\Internship\EcoComfort-Intelligent-Energy-Efficiency-for-Residential-Buildings>gcloud init
Welcome! This command will take you through the configuration of gcloud.

Settings from your current configuration [energy-efficiency] are:
accessibility:
  screen_reader: 'False'
core:
  account: sahusailendrakumar053@gmail.com
  disable_usage_reporting: 'True'

Pick configuration to use:
[1] Re-initialize this configuration [energy-efficiency] with new settings
[2] Create a new configuration
[3] Switch to and re-initialize existing configuration: [cement-s]
[4] Switch to and re-initialize existing configuration: [concrete-s]
[5] Switch to and re-initialize existing configuration: [default]
Please enter your numeric choice: 1
```

9. Select option 2 for new configuration and then write the name of the project that you created previously.



```
C:\Windows\System32\cmd.exe - gcloud init

E:\Learnings\iNeuron\Internship\EcoComfort-Intelligent-Energy-Efficiency-for-Residential-Buildings>gcloud init
Welcome! This command will take you through the configuration of gcloud.

Settings from your current configuration [energy-efficiency] are:
accessibility:
  screen_reader: 'False'
core:
  account: sahusailendrakumar053@gmail.com
  disable_usage_reporting: 'True'

Pick configuration to use:
[1] Re-initialize this configuration [energy-efficiency] with new settings
[2] Create a new configuration
[3] Switch to and re-initialize existing configuration: [cement-s]
[4] Switch to and re-initialize existing configuration: [concrete-s]
[5] Switch to and re-initialize existing configuration: [default]
Please enter your numeric choice: 2
```

10. Select the account that you want to use or login with another.
Then pick the project name to use.

```
C:\Windows\System32\cmd.exe - gcloud init

[3] Switch to and re-initialize existing configuration: [cement-s]
[4] Switch to and re-initialize existing configuration: [concrete-s]
[5] Switch to and re-initialize existing configuration: [default]
Please enter your numeric choice: 1

Your current configuration has been set to: [energy-efficiency]

You can skip diagnostics next time by using the following flag:
gcloud init --skip-diagnostics

Network diagnostic detects and fixes local network connection issues.
Checking network connection...done.
Reachability Check passed.
Network diagnostic passed (1/1 checks passed).

Choose the account you would like to use to perform operations for this configuration:
[1] sahushailendrakumar053@gmail.com
[2] Log in with a new account
Please enter your numeric choice: 1 ✓

You are logged in as: [sahushailendrakumar053@gmail.com].

Pick cloud project to use:
[1] concrete-s
[2] ecocomfort-energy-efficiency ✓
[3] flask-calc
[4] rock-atlas-393006
[5] Enter a project ID
[6] Create a new project
Please enter numeric choice or text value (must exactly match list item): 2
```

11. Run this command to deploy the project
`gcloud app deploy app.yaml --project <<project name>>`

```
C:\Windows\System32\cmd.exe

* Commands that require authentication will use sahushailendrakumar053@gmail.com by default
* Commands will reference project 'ecocomfort-energy-efficiency' by default
Run 'gcloud help config' to learn how to change individual settings

(This gcloud configuration is called [energy-efficiency]. You can create additional configurations if you work with multiple accounts and/or projects.
Run 'gcloud topic configurations' to learn more.

Some things to try next:

* Run 'gcloud --help' to see the Cloud Platform services you can interact with. And run 'gcloud help COMMAND' to get help on any gcloud command.
* Run 'gcloud topic --help' to learn about advanced features of the SDK like arg files and output formatting
* Run 'gcloud cheat-sheet' to see a roster of go-to 'gcloud' commands.

Updates are available for some Google Cloud CLI components. To install them,
please run:
$ gcloud components update

To take a quick anonymous survey, run:
$ gcloud survey

E:\Learnings\iNeuron\Internship\EcoComfort-Intelligent-Energy-Efficiency-for-Residential-Buildings>gcloud app deploy app.yaml --project ecocomfort-energy-efficiency
```


12. Now choose your region.

```
C:\Windows\System32\cmd.exe - gcloud app deploy app.yaml --project ecocomfort-energy-efficiency
cannot be changed. More information about regions is at
<https://cloud.google.com/appengine/docs/locations>.

Please choose the region where you want your App Engine application located:

[1] asia-east1      (supports standard and flexible)
[2] asia-east2      (supports standard and flexible and search_api)
[3] asia-northeast1 (supports standard and flexible and search_api)
[4] asia-northeast2 (supports standard and flexible and search_api)
[5] asia-northeast3 (supports standard and flexible and search_api)
[6] asia-south1      (supports standard and flexible and search_api)
[7] asia-southeast1 (supports standard and flexible)
[8] asia-southeast2 (supports standard and flexible and search_api)
[9] australia-southeast1 (supports standard and flexible and search_api)
[10] europe-central2 (supports standard and flexible)
[11] europe-west     (supports standard and flexible and search_api)
[12] europe-west2    (supports standard and flexible and search_api)
[13] europe-west3    (supports standard and flexible and search_api)
[14] europe-west6    (supports standard and flexible and search_api)
[15] northamerica-northeast1 (supports standard and flexible and search_api)
[16] southamerica-east1 (supports standard and flexible and search_api)
[17] us-central      (supports standard and flexible and search_api)
[18] us-east1         (supports standard and flexible and search_api)
[19] us-east4         (supports standard and flexible and search_api)
[20] us-west1         (supports standard and flexible)
[21] us-west2         (supports standard and flexible and search_api)
[22] us-west3         (supports standard and flexible and search_api)
[23] us-west4         (supports standard and flexible and search_api)
[24] cancel
Please enter your numeric choice: 6 ✓
```

13. It will complete the creation of app engine application. And will give us the target url to access the prediction page.

```
C:\Windows\System32\cmd.exe - gcloud app deploy app.yaml --project ecocomfort-energy-efficiency

[12] europe-west2    (supports standard and flexible and search_api)
[13] europe-west3    (supports standard and flexible and search_api)
[14] europe-west6    (supports standard and flexible and search_api)
[15] northamerica-northeast1 (supports standard and flexible and search_api)
[16] southamerica-east1 (supports standard and flexible and search_api)
[17] us-central      (supports standard and flexible and search_api)
[18] us-east1         (supports standard and flexible and search_api)
[19] us-east4         (supports standard and flexible and search_api)
[20] us-west1         (supports standard and flexible)
[21] us-west2         (supports standard and flexible and search_api)
[22] us-west3         (supports standard and flexible and search_api)
[23] us-west4         (supports standard and flexible and search_api)
[24] cancel
Please enter your numeric choice: 6

Creating App Engine application in project [ecocomfort-energy-efficiency] and region [asia-south1]....done.
Services to deploy:

descriptor:      [E:\Learnings\1Neuron\Internship\EcoComfort-Intelligent-Energy-Efficiency-for-Residential-B
buildings\app.yaml]
source:          [E:\Learnings\1Neuron\Internship\EcoComfort-Intelligent-Energy-Efficiency-for-Residential-B
buildings]
target project:  [ecocomfort-energy-efficiency]
target service:  [default]
target version:  [20230716t121540]
target url:      [https://ecocomfort-energy-efficiency.el.r.appspot.com] ✓
target service account: [ecocomfort-energy-efficiency@appspot.gserviceaccount.com]

Do you want to continue (Y/n)?
```


14. It will ask to continue uploading all the files into cloud.
Select Y to give it yes to continue.

```
C:\Windows\System32\cmd.exe - gcloud app deploy app.yaml --project ecocomfort-energy-efficiency

[16] southamerica-east1 (supports standard and flexible and search_api)
[17] us-central          (supports standard and flexible and search_api)
[18] us-east1            (supports standard and flexible and search_api)
[19] us-east4            (supports standard and flexible and search_api)
[20] us-west1            (supports standard and flexible)
[21] us-west2            (supports standard and flexible and search_api)
[22] us-west3            (supports standard and flexible and search_api)
[23] us-west4            (supports standard and flexible and search_api)
[24] cancel
Please enter your numeric choice: 6

Creating App Engine application in project [ecocomfort-energy-efficiency] and region [asia-south1]...done.
Services to deploy:

descriptor:      [E:\Learnings\iNeuron\Internship\EcoComfort-Intelligent-Energy-Efficiency-for-Residential-B
buildings\app.yaml]
source:          [E:\Learnings\iNeuron\Internship\EcoComfort-Intelligent-Energy-Efficiency-for-Residential-B
buildings]
target project:  [ecocomfort-energy-efficiency]
target service:  [default]
target version:  [20230716t121540]
target url:      [https://ecocomfort-energy-efficiency.el.r.appspot.com]
target service account: [ecocomfort-energy-efficiency@appspot.gserviceaccount.com]

Do you want to continue (Y/n)? y ✓
Beginning deployment of service [default]...
Created .gcloudignore file. See 'gcloud topic gcloudignore' for details.
```

15. After the upload is finished, copy the target url and paste it in the browser.
Or we can also go to the app engine then version and click on the given link to open it in the browser.

16. The prediction page will open. It has two options, custom file predict and default file predict.
Select default file predict to make predictions for the data kept in a default folder.

Prediction

Custom File Predict

Or

Default File Predict

Results

17. The prediction is completed, and prediction file is created.

Prediction

Custom File Predict

Or

Default File Predict

Results

"Prediction File created !!!

```
read-only file system: 'Prediction_Logs/Prediction_Log.txt'
```