**Energy Efficiency**

# Technical Design Document

**Version 1.0**

RESTRICTED DISTRIBUTION

The information is standard Company Confidential, but due to its sensitivity, it has restricted distribution and viewing within iNeuron.

Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| **Date Issued** | **Version** | **Description** | **Author** |
| 16th July 2020 | 1.0 | Initial Draft | Ruchik.k  nisikanta |

Contributors

The content of this document has been authored with the combined input of the following group of key individuals.

|  |  |
| --- | --- |
| **Name** | **Section Worked Upon** |
| Ruchik K | Initial Draft |
| Nisikanta Dash | Initial Draft |

Document Classification

|  |  |
| --- | --- |
| **Classification** | Company Confidential |
| **Definition** | Information is Group confidential and needs to be protected |
| **Context** | Where the loss of information confidentiality would result in significant harm to the interests of the organisation, financial loss, embarrassment or loss of information |

Contents

[1 Technical Design Document 2](#_gjdgxs)

[1. Introduction 5](#_30j0zll)

[High level objectives 5](#_1fob9te)

[2 Workflow Overall 6](#_3znysh7)

[Application Flow 6](#_2et92p0)

[Exception Scenarios Overall 6](#_tyjcwt)

[3 Workflow Data Ingestion and File Conversion 8](#_3dy6vkm)

[3.1 Technical solution design 10](#_1t3h5sf)

[3.2 Exceptions Scenarios 10](#_4d34og8)

[4 Stats Based EDA 11](#_2s8eyo1)

[4.1 Steps 11](#_17dp8vu)

[4.2 Technical solution design 11](#_3rdcrjn)

[4.3 Exceptions Scenarios Module Wise 11](#_26in1rg)

[5 Graph-Based EDA 12](#_lnxbz9)

[5.1 Technical solution design 12](#_35nkun2)

[5.2 Exceptions Scenarios Module Wise 13](#_1ksv4uv)

[6 Library Based Utils 14](#_44sinio)

[6.1 Technical solution design 14](#_2jxsxqh)

[6.2 Exceptions Scenarios Module Wise 14](#_z337ya)

[7 Data Transformers( Pre-processing steps) 15](#_3j2qqm3)

[7.1 Technical solution design 15](#_1y810tw)

[7.2 Exceptions Scenarios Module Wise 15](#_4i7ojhp)

[8 ML Model Selection 16](#_2xcytpi)

[8.1 Technical solution design 16](#_1ci93xb)

[8.2 Exceptions Scenarios Module Wise 16](#_3whwml4)

[9 Model Tuning and Optimization 17](#_2bn6wsx)

[9.1 Technical solution design 18](#_qsh70q)

[9.2 Exceptions Scenarios Module Wise 18](#_3as4poj)

[10 Testing Modules 19](#_1pxezwc)

[10.1 Technical solution design 19](#_49x2ik5)

[10.2 Exceptions Scenarios Module Wise 20](#_2p2csry)

[11 Prediction Pipeline 21](#_147n2zr)

[11.1 Technical solution design 22](#_3o7alnk)

[11.2 Exceptions Scenarios Module Wise 23](#_23ckvvd)

[12 Deployment Strategy 24](#_ihv636)

[12.1 Technical solution design 25](#_32hioqz)

[12.2 Exceptions Scenarios Module Wise 26](#_1hmsyys)

[13 Monitoring 27](#_41mghml)

[13.1 Technical solution design 27](#_2grqrue)

[13.2 Exceptions Scenarios Module Wise 27](#_vx1227)

[14 Logging 28](#_3fwokq0)

[14.1 Technical solution design 28](#_1v1yuxt)

[14.2 Exceptions Scenarios Module Wise 28](#_4f1mdlm)

[15 Hardware Requirements 29](#_2u6wntf)

[15.1 Requirements for model training 29](#_19c6y18)

[15.2 Requirements for model testing 29](#_3tbugp1)

# Introduction

The goal here is to build an end to end automated Machine Learning solution where the user will only give the data and select the type of problem, and the result will be the best performing hyper tuned Machine Learning model. The user will also get privileges to choose the deployment options.

This project shall be delivered in two phases:

Phase 1: All the functionalities with PyPi packages.

Phase2: Integration of UI to all the functionalities.

The technical design document gives a design blueprint of the Enerygy Efficiency project. This document communicates the technical details of the solution proposed.

In addition, this document also captures the different workflows involved to build the solution, exceptions in the workflows and any assumptions that have been considered.

Once agreed as the basis for the building of the project, the flowchart and assumptions will be used as a platform from which the solution will be designed.

Changes to this business process may constitute a request for change and will be subject to the agreed agility program change procedures.

**Note: All the code will be written in python version 3.7**

## High level objectives

The high-level objectives are:

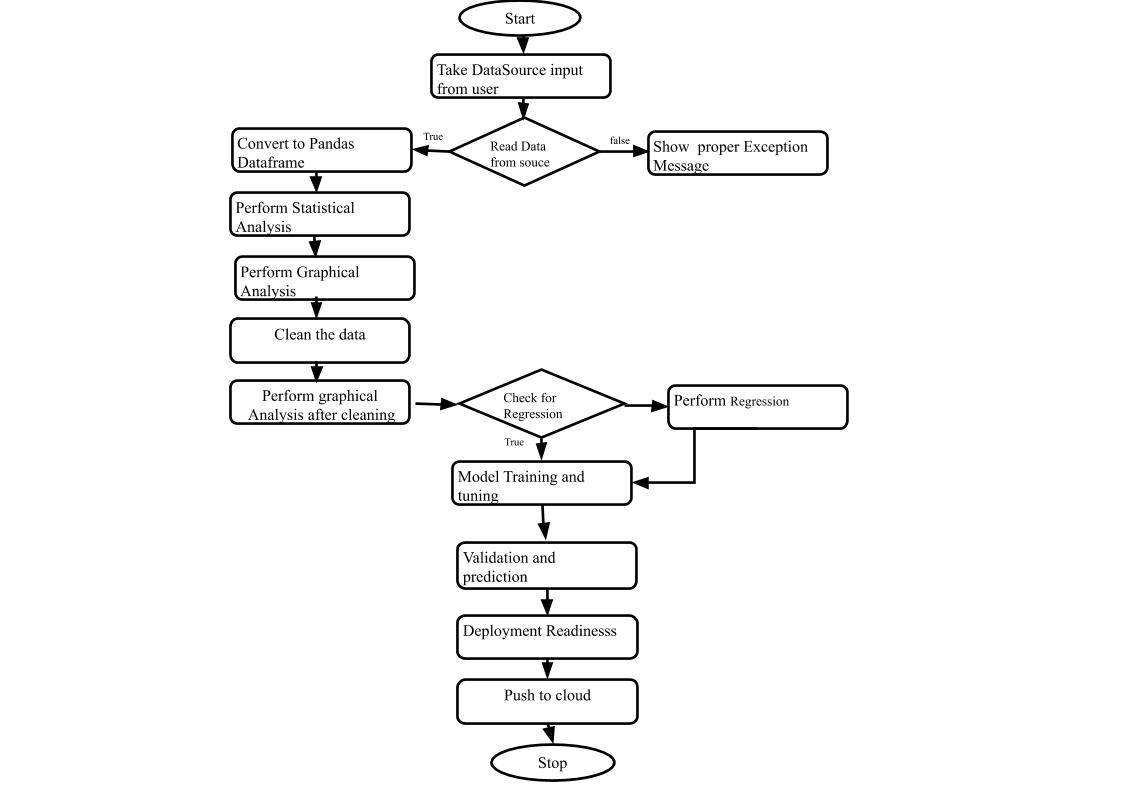
1. Enable reading/loading of data from the various sources and convert them into pandas dataframe(details mentioned in the Data Ingestion Section).
2. Enable reading various file formats and convert them into pandas dataframe(details mentioned in the Data Ingestion Section).
3. Give user the option to specify feature and target columns.
4. Give user the option to select the problem type, viz. Regression, Classification (include anomaly detection), Clustering or Time Series.
5. Perform statistical analytics of the data and prepare a table for the analysis and show it on screen.
6. Perform graphical analysis for the data and Showcase the results (graphs) on the screen.
7. Perform data cleaning operation with all the steps required and showcase a report on screen.
8. After data cleaning showcase the graphical analysis once again for comparison.
9. Check whether clustering is required or not.
10. Choose the appropriate ML model for training.
11. Perform model Tuning.
12. Create a list of top 3 models and show multiple metrics for them.
13. Give option for prediction.
14. Give options for docker container creation.
15. Give option for automatic cloud deployment.

**Phase 1:** Create Pypi packages

**Phase 2:** Create UI

# Workflow Overall

## Application Flow



## Exception Scenarios Overall

|  |  |  |
| --- | --- | --- |
| **Step** | **Exception** | **Mitigation** |
| User gives Wrong Data Source | Give proper error message | Ask the user to re-enter the details |
| User gives corrupted data | Give proper error message |  |
| If the cluster contains only one class | No error message required | Handle this exception internally. User doesn’t know. |
| Deployment credentials are wrong | Give proper error message | Ask for the details to be entered again |

# Workflow Data Ingestion and File Conversion

**Data Sources:**

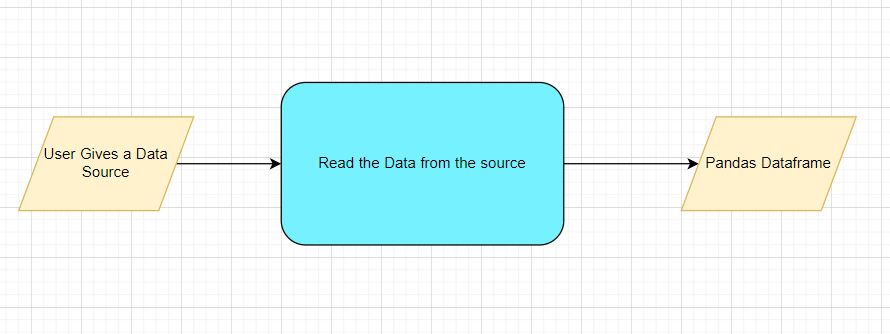
**Phase 1:**

|  |  |
| --- | --- |
| Data Connector Utils | File Conversion Utils |
| [Microsoft Access](https://help.tableau.com/current/pro/desktop/en-us/examples_access.htm) | CSV |
| [Spatial File](https://help.tableau.com/current/pro/desktop/en-us/examples_spatial_files.htm) | JSON |
| [Statistical File](https://help.tableau.com/current/pro/desktop/en-us/examples_statfile.htm) | HTML |
| [Tableau Server or Tableau Online](https://help.tableau.com/current/pro/desktop/en-us/examples_tableauserver.htm) | Excel files |
| [Actian Matrix](https://help.tableau.com/current/pro/desktop/en-us/examples_actianmatrix.htm) | OpenDocument Spreadsheets |
| [Actian Vectorwise](https://help.tableau.com/current/pro/desktop/en-us/examples_vectorwise.htm) | Binary Excel (.xlsb) files |
| [Alibaba AnalyticDB for MySQL](https://help.tableau.com/current/pro/desktop/en-us/examples_alibaba_analyticdb.htm) | Clipboard |
| [Alibaba Data Lake Analytics](https://help.tableau.com/current/pro/desktop/en-us/examples_alibaba_data_lake_analytics.htm) | Pickling |
| [Alibaba MaxCompute](https://help.tableau.com/current/pro/desktop/en-us/examples_alibaba_maxcompute.htm) | msgpack |
| [Amazon Athena](https://help.tableau.com/current/pro/desktop/en-us/examples_amazonathena.htm) | HDF5 (PyTables) |
| [Amazon Aurora for MySQL](https://help.tableau.com/current/pro/desktop/en-us/examples_amazonaurora.htm) | Feather |
| [Amazon EMR Hadoop Hive](https://help.tableau.com/current/pro/desktop/en-us/examples_amazonemr.htm) | Parquet |
| [Amazon Redshift](https://help.tableau.com/current/pro/desktop/en-us/examples_amazonredshift.htm) | ORC |
| [Anaplan](https://help.tableau.com/current/pro/desktop/en-us/examples_anaplan.htm) | Google BigQuery |
| [Apache Drill](https://help.tableau.com/current/pro/desktop/en-us/examples_apachedrill.htm) | Stata format |
| [Aster Database](https://help.tableau.com/current/pro/desktop/en-us/examples_asterdata.htm) | SAS formats |
| [Azure SQL Synapse Analytics](https://help.tableau.com/current/pro/desktop/en-us/examples_azure_sql_dw.htm) | SPSS formats |
| [Box](https://help.tableau.com/current/pro/desktop/en-us/examples_box.htm) | Other file formats |
| [Cloudera Hadoop](https://help.tableau.com/current/pro/desktop/en-us/examples_hadoop.htm) | Performance considerations |
| [Databricks](https://help.tableau.com/current/pro/desktop/en-us/examples_databricks.htm) |  |
| [Denodo](https://help.tableau.com/current/pro/desktop/en-us/examples_denodo.htm) |  |
| [Dropbox](https://help.tableau.com/current/pro/desktop/en-us/examples_dropbox.htm) |  |
| [Esri ArcGIS Server](https://help.tableau.com/current/pro/desktop/en-us/examples_esri.htm) |  |
| [Exasol](https://help.tableau.com/current/pro/desktop/en-us/examples_exasolution.htm) |  |
| [Firebird 3](https://help.tableau.com/current/pro/desktop/en-us/examples_firebird.htm) |  |
| [Google Ads](https://help.tableau.com/current/pro/desktop/en-us/examples_googleads.htm) |  |
| [Google Analytics](https://help.tableau.com/current/pro/desktop/en-us/examples_googleanalytics.htm) |  |
| [Google BigQuery](https://help.tableau.com/current/pro/desktop/en-us/examples_googlebigquery.htm) |  |
| [Google Cloud SQL](https://help.tableau.com/current/pro/desktop/en-us/examples_googlecloudsql.htm) |  |
| [Google Drive](https://help.tableau.com/current/pro/desktop/en-us/examples_googledrive.htm) |  |
| [Google Sheets](https://help.tableau.com/current/pro/desktop/en-us/examples_googlesheets.htm) |  |
| [Hortonworks Hadoop Hive](https://help.tableau.com/current/pro/desktop/en-us/examples_hortonworkshadoop.htm) |  |
| [IBM BigInsights](https://help.tableau.com/current/pro/desktop/en-us/examples_biginsights.htm) |  |
| [IBM DB2](https://help.tableau.com/current/pro/desktop/en-us/examples_db2.htm) |  |
| [IBM PDA (Netezza)](https://help.tableau.com/current/pro/desktop/en-us/examples_netezza.htm) |  |
| [Impala](https://help.tableau.com/current/pro/desktop/en-us/examples_impala.htm) |  |
| [Intuit QuickBooks Online](https://help.tableau.com/current/pro/desktop/en-us/examples_quickbooksonline.htm) |  |
| [Kognitio](https://help.tableau.com/current/pro/desktop/en-us/examples_kognitio.htm) |  |
| [Kyvos](https://help.tableau.com/current/pro/desktop/en-us/examples_kyvos.htm) |  |
| [LinkedIn Sales Navigator](https://help.tableau.com/current/pro/desktop/en-us/examples_linkedin_sales_navigator.htm) |  |
| [MapR Hadoop Hive](https://help.tableau.com/current/pro/desktop/en-us/examples_maprhadoop.htm) |  |
| [MariaDB](https://help.tableau.com/current/pro/desktop/en-us/examples_mariadb.htm) |  |
| [Marketo](https://help.tableau.com/current/pro/desktop/en-us/examples_marketo.htm) |  |
| [MarkLogic](https://help.tableau.com/current/pro/desktop/en-us/examples_marklogic.htm) |  |
| [MemSQL](https://help.tableau.com/current/pro/desktop/en-us/examples_memsql.htm) |  |
| [Microsoft Analysis Services](https://help.tableau.com/current/pro/desktop/en-us/examples_msas.htm) |  |
| [Microsoft PowerPivot](https://help.tableau.com/current/pro/desktop/en-us/examples_powerpivot.htm) |  |
| [Microsoft SQL Server](https://help.tableau.com/current/pro/desktop/en-us/examples_sqlserver.htm) |  |
| [MonetDB](https://help.tableau.com/current/pro/desktop/en-us/examples_monetdb.htm) |  |
| [MongoDB BI Connector](https://help.tableau.com/current/pro/desktop/en-us/examples_mongodb.htm) |  |
| [MySQL](https://help.tableau.com/current/pro/desktop/en-us/examples_mysql.htm) |  |
| [OData](https://help.tableau.com/current/pro/desktop/en-us/examples_odata.htm) |  |
| [OneDrive](https://help.tableau.com/current/pro/desktop/en-us/examples_onedrive.htm) |  |
| [Oracle](https://help.tableau.com/current/pro/desktop/en-us/examples_oracle.htm) |  |
| [Oracle Eloqua](https://help.tableau.com/current/pro/desktop/en-us/examples_eloqua.htm) |  |
| [Oracle Essbase](https://help.tableau.com/current/pro/desktop/en-us/examples_essbase.htm) |  |
| [Pivotal Greenplum](https://help.tableau.com/current/pro/desktop/en-us/examples_greenplum.htm) |  |
| [PostgreSQL](https://help.tableau.com/current/pro/desktop/en-us/examples_postgresql.htm) |  |
| [Presto](https://help.tableau.com/current/pro/desktop/en-us/examples_presto.htm) |  |
| [Progress OpenEdge](https://help.tableau.com/current/pro/desktop/en-us/examples_progress.htm) |  |
| [Qubole Presto](https://help.tableau.com/current/pro/desktop/en-us/examples_qubole.htm) |  |
| [Salesforce](https://help.tableau.com/current/pro/desktop/en-us/examples_salesforce.htm) |  |
| [Splunk](https://help.tableau.com/current/pro/desktop/en-us/examples_splunk.htm) |  |
| [SAP HANA](https://help.tableau.com/current/pro/desktop/en-us/examples_saphana.htm) |  |
| [SAP NetWeaver Business Warehouse](https://help.tableau.com/current/pro/desktop/en-us/examples_sapbw.htm) |  |
| [SAP Sybase ASE](https://help.tableau.com/current/pro/desktop/en-us/examples_sybasease.htm) |  |
| [SAP Sybase IQ](https://help.tableau.com/current/pro/desktop/en-us/examples_sybaseiq.htm) |  |
| [ServiceNow ITSM](https://help.tableau.com/current/pro/desktop/en-us/examples_servicenow.htm) |  |
| [SharePoint Lists](https://help.tableau.com/current/pro/desktop/en-us/examples_sharepoint_lists.htm) |  |
| [Snowflake](https://help.tableau.com/current/pro/desktop/en-us/examples_snowflake.htm) |  |
| [Spark SQL](https://help.tableau.com/current/pro/desktop/en-us/examples_sparksql.htm) |  |
|  |  |
|  |  |
|  |  |
|  |  |
| [Connector Plugin](https://help.tableau.com/current/pro/desktop/en-us/examples_connector_sdk.htm) |  |
| [Web Data Connector](https://help.tableau.com/current/pro/desktop/en-us/examples_web_data_connector.htm) |  |
| [Other Databases (JDBC)](https://help.tableau.com/current/pro/desktop/en-us/examples_otherdatabases_jdbc.htm) |  |
| [Other Databases (ODBC)](https://help.tableau.com/current/pro/desktop/en-us/examples_otherdatabases.htm) |  |

**Phase 2:**

|  |  |
| --- | --- |
| Data Connector Utils | File Conversion Utils |
| [Spatial File](https://help.tableau.com/current/pro/desktop/en-us/examples_spatial_files.htm) | OpenDocument Spreadsheets |
| [Statistical File](https://help.tableau.com/current/pro/desktop/en-us/examples_statfile.htm) |  |
| [Tableau Server or Tableau Online](https://help.tableau.com/current/pro/desktop/en-us/examples_tableauserver.htm) |  |
| [Actian Matrix](https://help.tableau.com/current/pro/desktop/en-us/examples_actianmatrix.htm) |  |
| [Teradata OLAP Connector](https://help.tableau.com/current/pro/desktop/en-us/examples_teradata_olap.htm) |  |
| [TIBCO Data Virtualization (Cisco Information Server)](https://help.tableau.com/current/pro/desktop/en-us/examples_ciscoinfoserver.htm) |  |
| [Vertica](https://help.tableau.com/current/pro/desktop/en-us/examples_vertica.htm) |  |
| [Teradata](https://help.tableau.com/current/pro/desktop/en-us/examples_teradata.htm) |  |

## Technical solution design



## Method Definitions

|  |  |  |
| --- | --- | --- |
| **Class Name** | **DataGetter** |  |
| Method Name | getdata |  |
|  | Method Description | This method will be used to read data from a csv file. |
|  | Input parameter names | self |
|  | Input Parameter Description |  |
|  | Out put | A pandas Dataframe |
|  | Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |
| Method Name | validationFileNameRaw |  |
|  | Method Description | This method will be used to validate the filename |
|  | Input parameter names | self,regex |
|  | Input Parameter Description | It will check the filename as inputfile.csv |
|  | ouptput | Check the valid file name |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |
|  | ouptput | A Pandas Dataframe |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |
|  | ouptput | A Pandas Dataframe |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |

## Exceptions Scenarios

|  |  |  |
| --- | --- | --- |
| **Step** | **Exception** | **Mitigation** |
| User gives Wrong Data Source | Give proper error message | Ask the user to re-enter the details |
| User gives corrupted data | Give proper error message |  |

# Data Profiling

After reading the data, automatically the following details should be shown:

1. The number of rows
2. The number of columns
3. Number of missing values per column and their percentage
4. Total missing values and it’s percentage
5. Number of categorical columns and their list
6. Number of numerical columns and their list
7. Number of duplicate rows
8. Number of columns with zero standard deviation and their list
9. Size occupied in RAM

## Method Definition

|  |  |  |
| --- | --- | --- |
| **Class Name** | **DataProfiler** |  |
| Method Name | ProfileReport() |  |
|  | Method Description | This method will be used to give various insight about data. |
|  | Input parameter names | self, dataframe |
|  | Input Parameter Description | dataframe: the input data just loaded from source |
|  | ouptput | 1. The number of rows 2. The number of columns 3. Number of missing values per column and their percentage 4. Total missing values and it’s percentage 5. Number of categorical columns and their list 6. Number of numerical columns and their list 7. Number of duplicate rows 8. Number of columns with zero standard deviation and their list 9. Size occupied in RAM |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |

# Stats Based EDA

## Steps

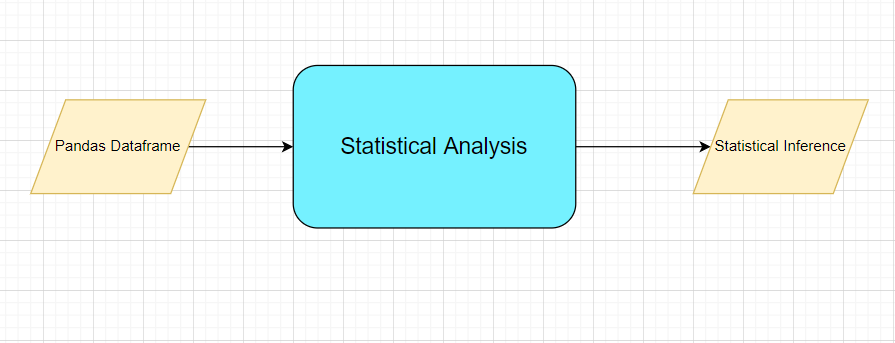
**MVP**

Correlation

**Phase1:**

Column contributions/ importance

## Technical solution design



## Exceptions Scenarios Module Wise

|  |  |  |
| --- | --- | --- |
| **Step** | **Exception** | **Mitigation** |
| Column has mixed values(Integer & number) | Give proper error message | Ask the user to correct the data. |
| Not all values are numbers | Handle Internally | Convert categorical to numerical values |

# Graph-Based EDA

Create the following graphs:

**MVP:**

Correlation Heatmaps

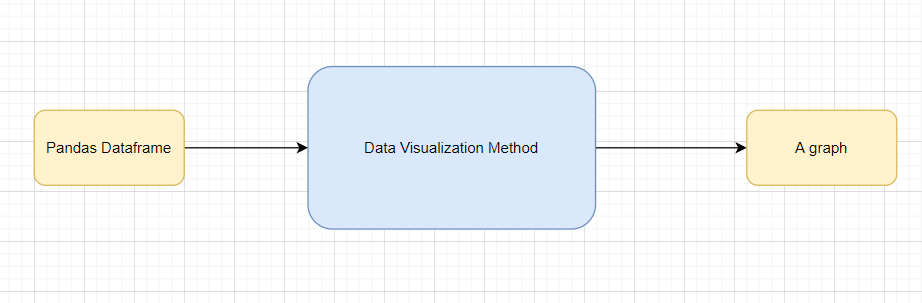
Check for balance/imbalance

**Phase1:**

Boxplot for outliers

Scatterplot

## Technical solution design



## Method Definitions

|  |  |  |
| --- | --- | --- |
| **Class Name** | **DataVisualization** |  |
| Method Name | getdata |  |
|  | Method Description | This method will be used to read data from a csv file or a flat file |
|  | Input parameter names | self |
|  | Input Parameter Description | file\_name: name of the file to be read  header: Row number(s) to be used as column names  names : array-like, optional  List of column names to use. If file contains no header row, then you  should explicitly pass ``header=None``.  Use\_cols: To load a subset of columns  Separator: Delimiter to use |
|  | ouptput | A pandas Dataframe |
|  | On Exception | Write the exception in the log file.  Raise an exception with the appropriate error message |

## Exceptions Scenarios Module Wise

|  |  |  |
| --- | --- | --- |
| **Step** | **Exception** | **Mitigation** |
| Wrong input to the methods | Handle Internally | Code should never give a wrong input |

# Library Based Utils

## Technical solution design

## Exceptions Scenarios Module Wise

## Data Transformers( Pre-processing steps)

**MVP:**

Null value handling

Normalisation

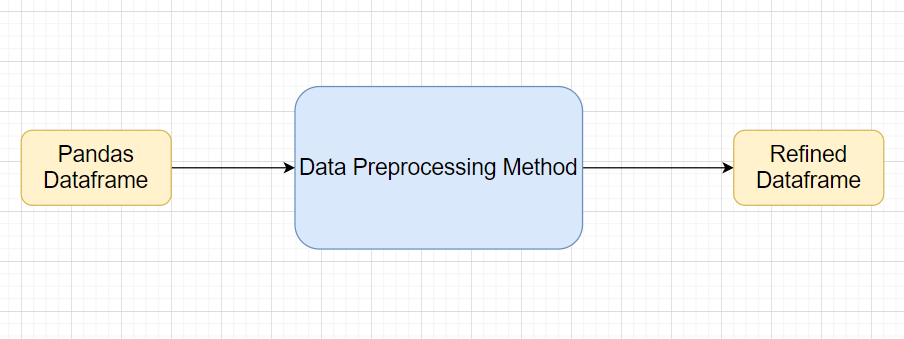
**Phase1:**

Outlier detection

Data Scaling/ Normalisation

Feature Selection: <https://scikit-learn.org/stable/auto_examples/index.html#feature-selection>

## Technical solution design



## Method Definitions

## Exceptions Scenarios Module Wise

|  |  |  |
| --- | --- | --- |
| **Step** | **Exception** | **Mitigation** |
| Wrong parameters passed to the methods | Handle Internally | Code should never give a wrong input |

# ML Model Selection

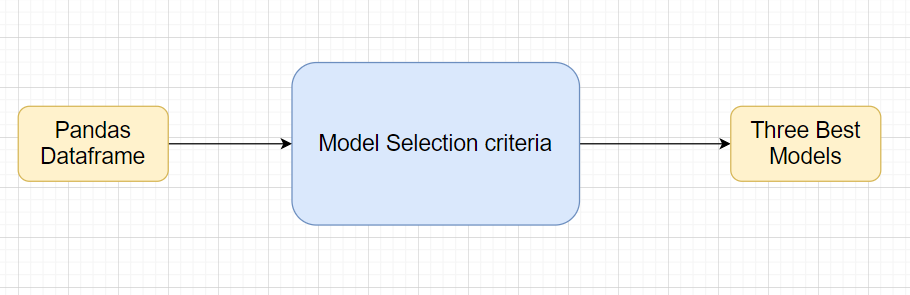
**MVP:**

3 Models—Linear Regression, Random Forest , Decision tree, XGBoost

**Phase1:**

Model Selection criteria

## Technical solution design



## Exceptions Scenarios Module Wise

|  |  |  |
| --- | --- | --- |
| **Step** | **Exception** | **Mitigation** |
| Wrong parameters passed to the methods | Handle Internally | Code should never give a wrong input |

# Model Tuning and Optimization

**Note:** The data should have been divided into train and validation set before this.

Methods for hyper tuning all kinds of models.

**Regression:**

Linear Regression

Decision Tree

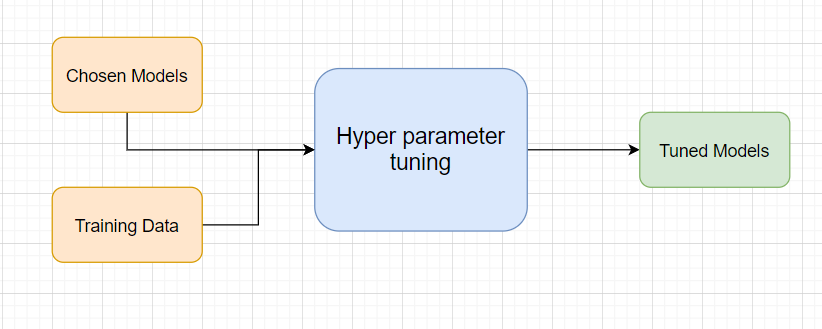
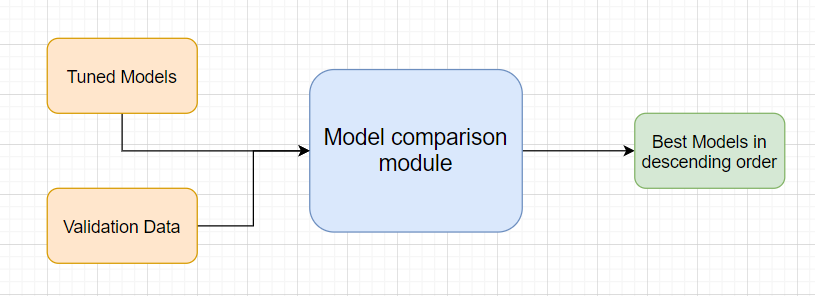
Random Forest

XG Boost

Model selection criteria:

MSE, RMSE, R squared, adjusted R squared : we have choose R squared

## Technical solution design

1. 
2. 

## Method Definitions

## Exceptions Scenarios Module Wise

|  |  |  |
| --- | --- | --- |
| **Step** | **Exception** | **Mitigation** |
|  |  |  |

# Testing Modules

Divide the training data itself into train and test sets

Use test data to have tests run on the three best models

Give the test report

1. R2 Score
2. Adjusted R2 score
3. MSE
4. Accuracy
5. Precision
6. Recall
7. F Beta
8. Cluster Purity
9. Silhouette score

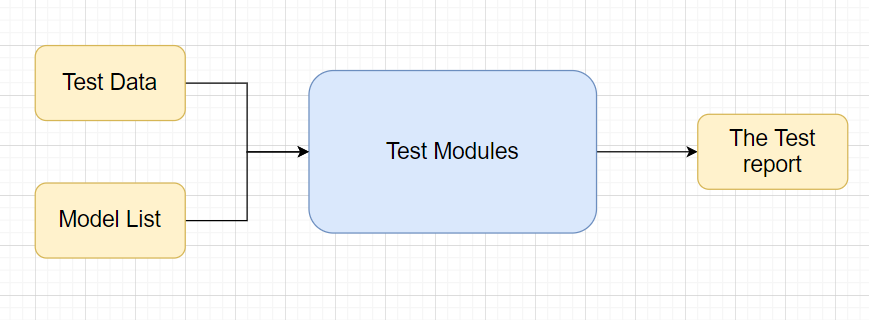
**Phase 2**

AIC

BIC

**Note**: Save the best model after validation is completed.

## Technical solution design



## Exceptions Scenarios Module Wise

|  |  |  |
| --- | --- | --- |
| **Step** | **Exception** | **Mitigation** |
| Number of Parameters do not match | Handle internally | Check the test data creation and verify the columns |
| Only once class present in test data | Handle Internally |  |

# Prediction Pipeline

Use the existing data read modules

Use the existing pre-processing module

Load the model into memory

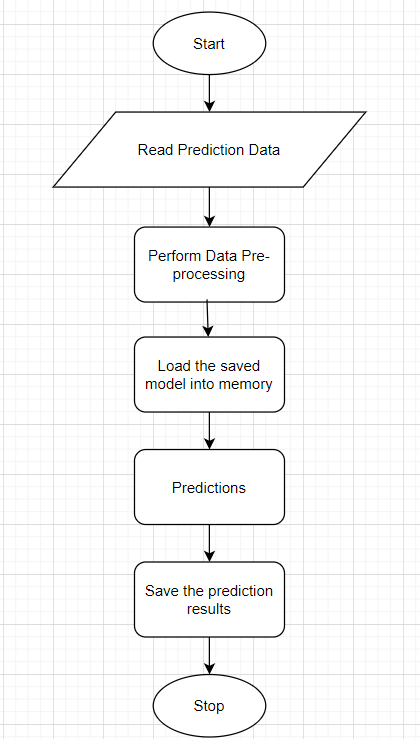
Do predictions

Store prediction results(show sample predictions)

Phase 2:

UI for predictions

## Technical solution design



## Exceptions Scenarios Module Wise

|  |  |  |
| --- | --- | --- |
| **Step** | **Exception** | **Mitigation** |
| Columns don’t match in training and Prediction data | Show error message | The user enters the correct data |
|  |  |  |

# Deployment Strategy

Take the cloud name as input

Prepare the metadata files based on cloud

Phase 2:

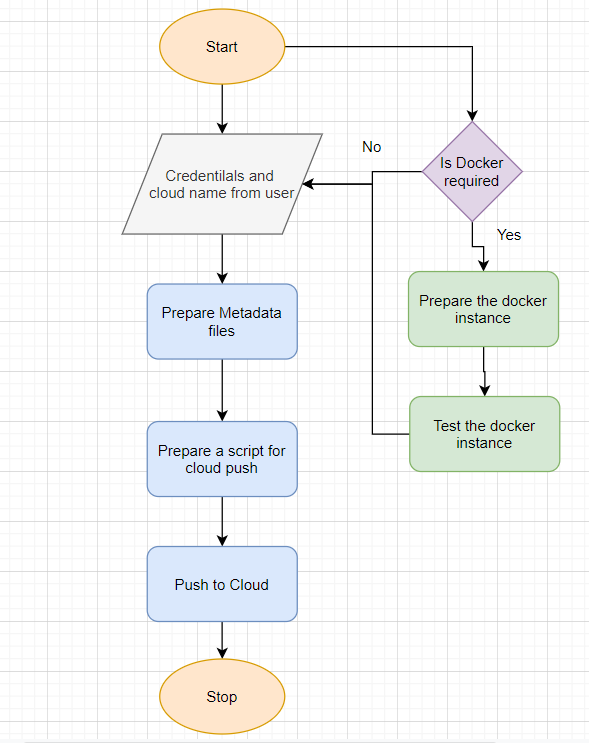
Accept the user credentials

Prepare a script file to push changes

Docker instance

Push of the docker instance to cloud

## Technical solution design



## Exceptions Scenarios Module Wise

|  |  |  |
| --- | --- | --- |
| **Step** | **Exception** | **Mitigation** |
| Wrong Cloud credentials | Show error message | The user enters the correct data |
| Docker instance not working | Show error message | Fix the error |
| Cloud push failed | Show the error | Make corrections to the metadata  files |
| Cloud app not starting |  | Ask the user for cloud logs for debugging |

# Monitoring

Phase 2

No. Of predictions for individual classes

No. of predictions (per day, per hour, per week etc.)

No. of hits

Training data size (number of rows)

Time spent in training

Failures

## Technical solution design



## Exceptions Scenarios Module Wise

|  |  |  |
| --- | --- | --- |
| **Step** | **Exception** | **Mitigation** |
|  |  |  |

# Logging

Separate Folder for logs

Entry to the methods

Exit from the methods with success/ failure message

Error message Logging

Model comparisons

**Phase 2:**

Options for Log Publish

## Technical solution design



## Common Logging Framework Code

|  |  |
| --- | --- |
| Class Name | App Logger |
| Method Name | log |
| Method Description | This method will be used for logging all the information to the file. |
| Input parameter names | self,file\_object, log\_message |
| Input Parameter Description | file\_object: the file where the logs will be written  log\_message: the message to be logged |
| ouptput | A log file with messages |

# from datetime import datetime class App\_Logger: def \_\_init\_\_(self): pass def log(self, file\_object, log\_message):“””This method will be used for logging all the information to the file.””” self.now = datetime.now() self.date = self.now.date() self.current\_time = self.now.strftime("%H:%M:%S") file\_object.write( str(self.date) + "/" + str(self.current\_time) + "\t\t" + log\_message +"\n")

## Exceptions Scenarios Module Wise

Ideally, the logging should never fail.

# Hardware Requirements

## Requirements for model training

The minimum configuration should be:

* 4 GB RAM
* 10 GB of Hard Disk Space
* Intel Core i5 Processor

## Requirements for model testing

The minimum configuration should be:

* 4 GB RAM
* 10 GB of Hard Disk Space
* Intel Core i5 Processor

**Deployment Steps :**

1. login to AWs Ec2 Instance

2.click on EC2 dashboard

3.click on lunch instance

4. select Ubuntu Server 18.04 LTS (HVM), SSD Volume Type (Free Tier)

5.Choose an Instance Type t2 small

6.Configure Instance Details as it is click on next:add Storage (We have provided 30 GB)

7.click on Add tags

8. click on add tag name ( provide value name)

9.configure security group and click on add rule ..(We have open the port 80 type http)

10. create a new keypair and keep it in safe place.

11. choose our new key pair and click on launch instance

12. wait for some time to our instance has been ready

**Steps for generate ppk file**

**1**. open the putty gen and load the file

2. save the file as ppk

1. Steps for connect to ec2 instance and configuration
2. sudo apt-get update
3. We have checked for pip3 we have not found so we run the below command
4. sudo apt install python3-pip
5. we install flask pip3 install flask
6. we install nginx ->sudo apt-get install nginx
7. We install gunicorn =>sudo apt-get install gunicorn3

**create a project directory through command**

1. sudo mkdir EnergyEfficiency

**move your file from local system to Ubuntu (Aws ec2 instance) and do the necessary changes**

1. install requirements.txt ->pip3 install -r requirements.txt
2. once its successfully installed
3. go to nginx sites-enabled folder: cd /etc/nginx/sites-enabled/
4. run the below command

sudo vim Energy

change the below configuratrion

server {

listen 80;

server\_name 18.237.157.208;

location / {

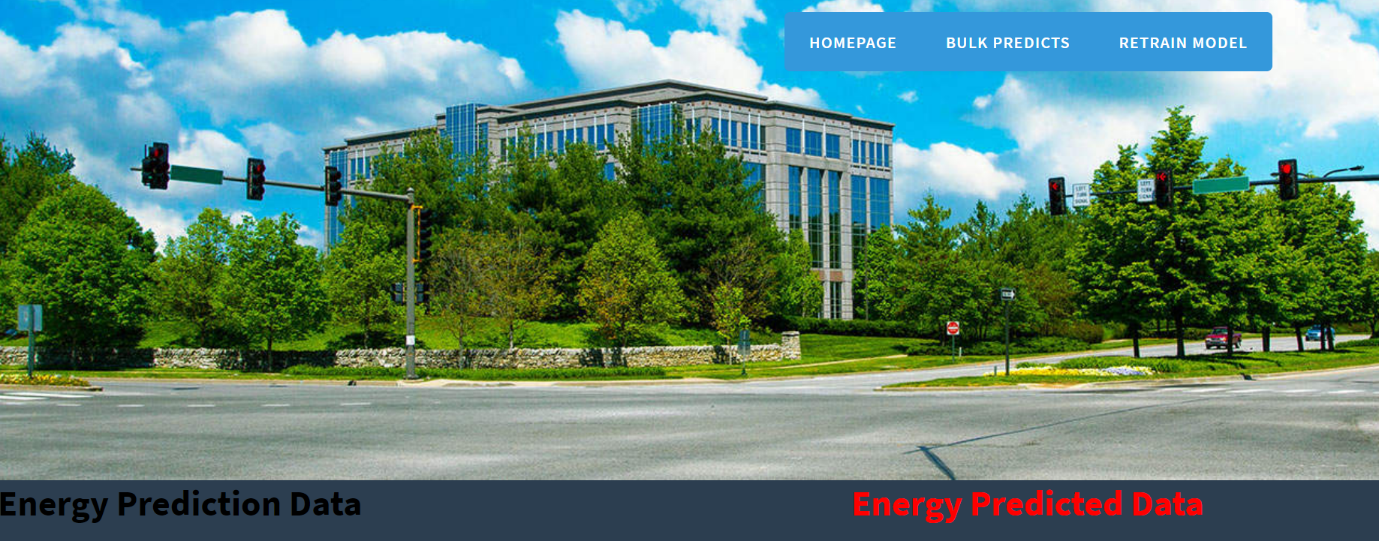
proxy\_pass http://127.0.0.1:8000;

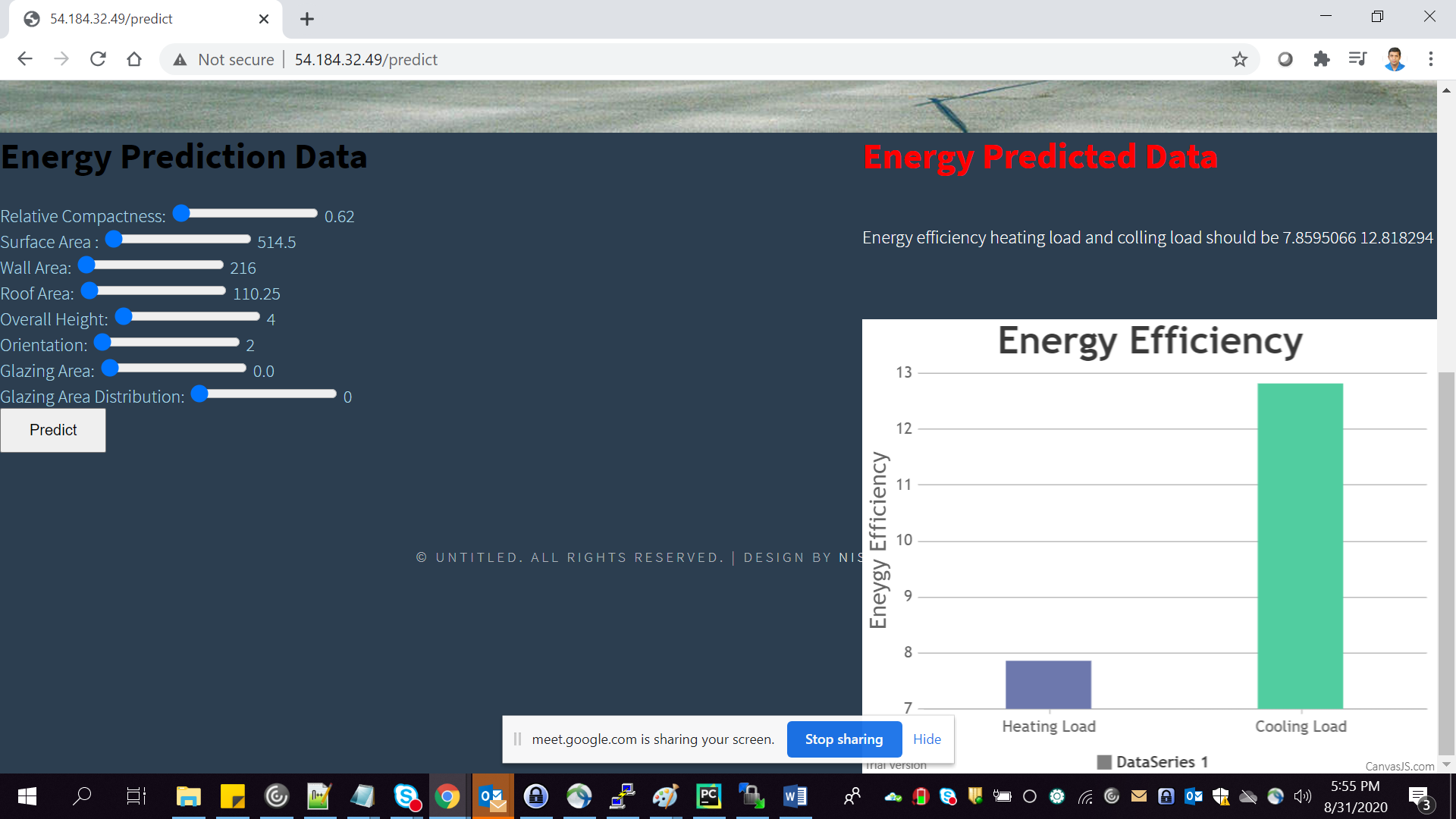
}

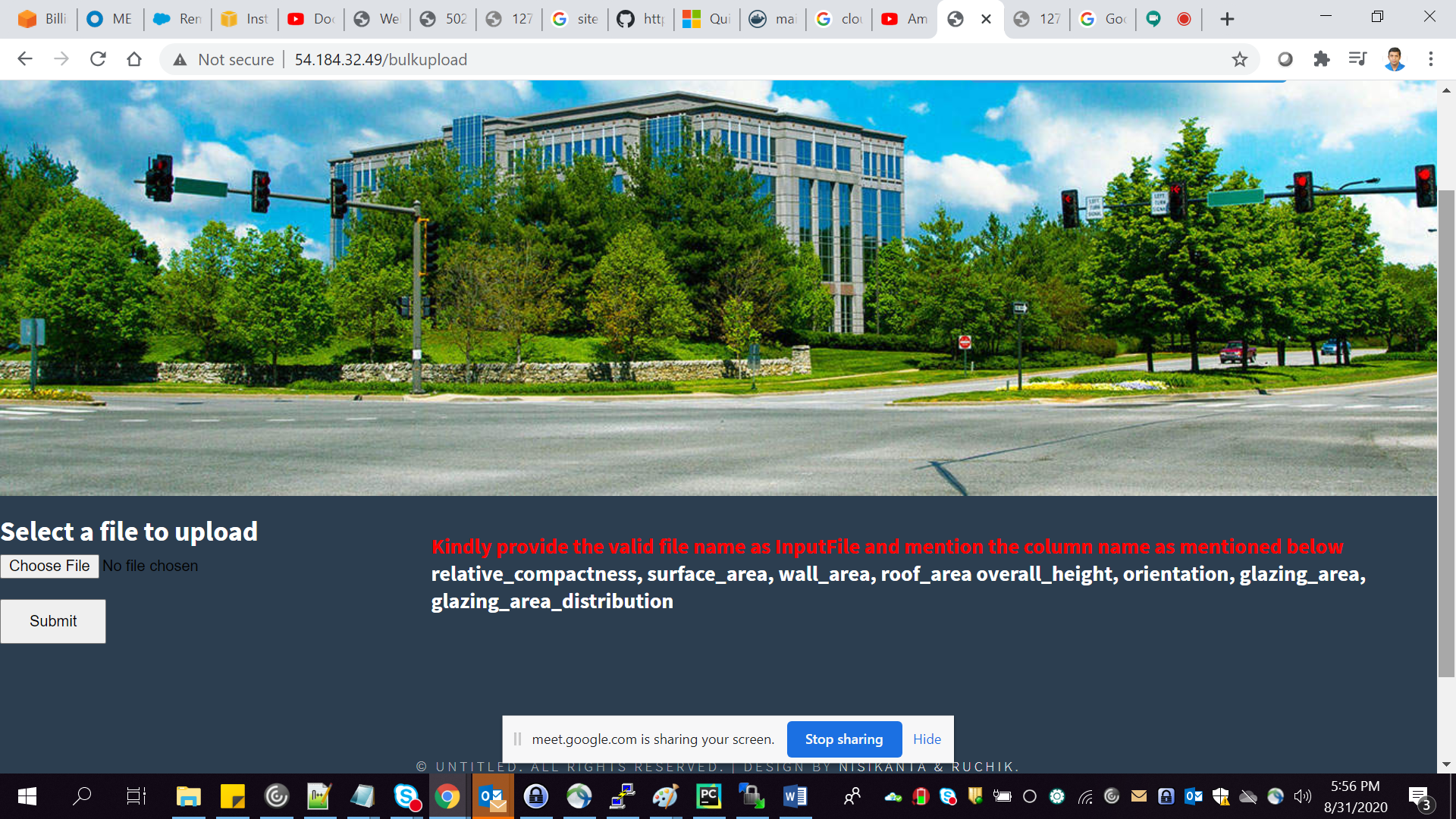
}

1. save the file and exit
2. restart the nginx service : sudo service nginx restart
3. go to the project directory: and run :- gunicorn3 main:app
4. Access the pubic IP of your instance it will run the application

Site Scree shots







**Step for create a gunicorn as a service**

1. home directory : cd
2. move to :- cd /etc/systemd/system/
3. run the command :- sudo vim gunicorn3\_energy.service
4. [Unit]
5. Description= Gunicorn Energy Service
6. After=network.target
7. [Service]
8. User=ubuntu
9. Group=www-data
10. WorkingDirectory=/home/ubuntu/EnergyEfficiency
11. ExecStart=/usr/bin/gunicorn3 --workers 3 --bind unix:EnergyEfficiency.sock -m 007 main:app
12. Save this file and exit
13. run the command : sudo systemctl daemon-reload
14. run the command : sudo service gunicorn3\_energy start
15. run the command : sudo service gunicorn3\_energy status
16. go to nginx sites-enabled: Sudo vim Energy
17. change the proxy pass : proxy\_pass http://unix:/home/ubuntu/EnergyEfficiency.sock
18. save the file and exit
19. run the command : sudo service nginx restart
20. run the command : sudo service gunicorn3\_energy start

Dockerization:

**Steps for install the docker**

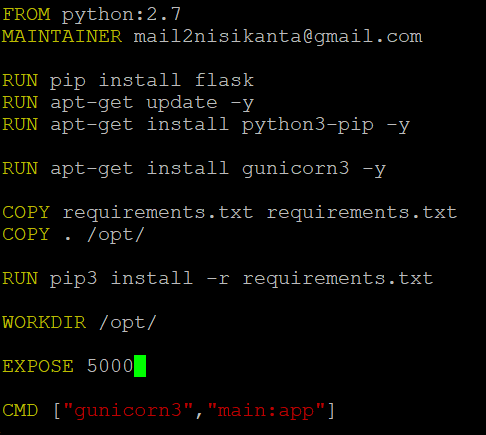
1. 1st check with command in root directory : sudo docker install

If you got the error message command not found

Then run the command: sudo apt install docker.io

Run the command: sudo snap install docker

1. Create the docker file sudo touch Dockerfile
2. Sudo vim Dockerfile
3. Make the below changes on the docker file



Build the docker image by the below command: sudo docker build -t ec2energyimage3 .

Run the docker image : sudo docker run ec2energyimage:latest

Login to the docker hub

After that we are able to login successfully

Sudo docker images: find your dockername

Then add tag to your docker : sudo docker tag ec2energyimage1 mail2nisikanta/ec2energyimage1

To push to docker image to docker hub: sudo docker push mail2nisikanta/ec2energyimage1

To pull your image in new instance

Create a new aws ec2 instance

Install docker

Login to to docker hub

Pull the command

docker pull mail2nisikanta/ec2energyimage1:latest

Run the docker : sudo docker run ec2energyimage:latest

huddles:

Aws deployment

Dockerization