Shell.ai 2023 Hackathon for Sustainable and Affordable Energy Agricultural Waste Challenge

(TEAM PREMIER)

Meet the Team

Sodiq Babawale

Industrial and Production Engineering, UI.

Machine Learning Engineer, Operations Research Analyst.

Femi Olashupo

Agricultural and Environmental Engineering, UI.

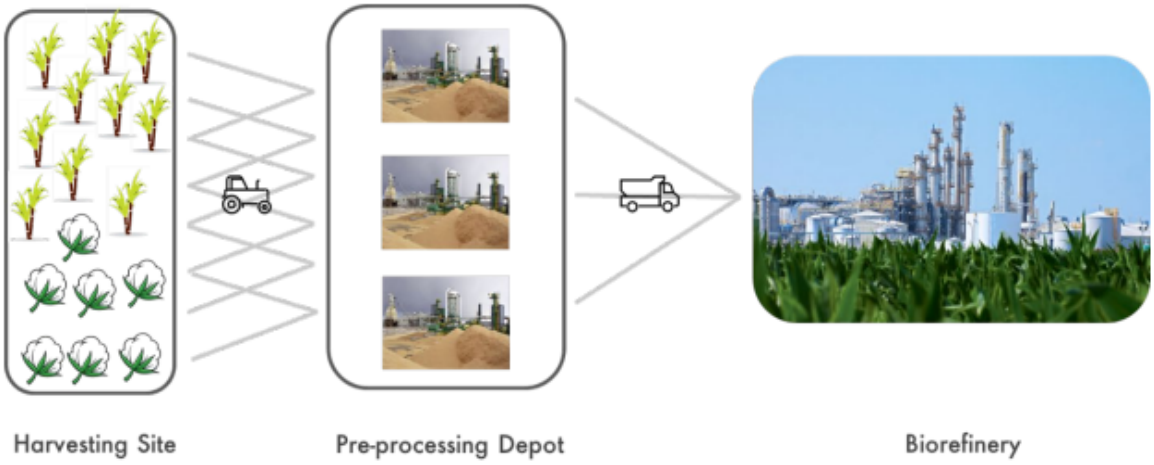
Machine Learning Engineer, Renewable Energy Researcher

Israel Odeajo

Computer Science, UI.

Machine Learning Engineer, Data Analyst.

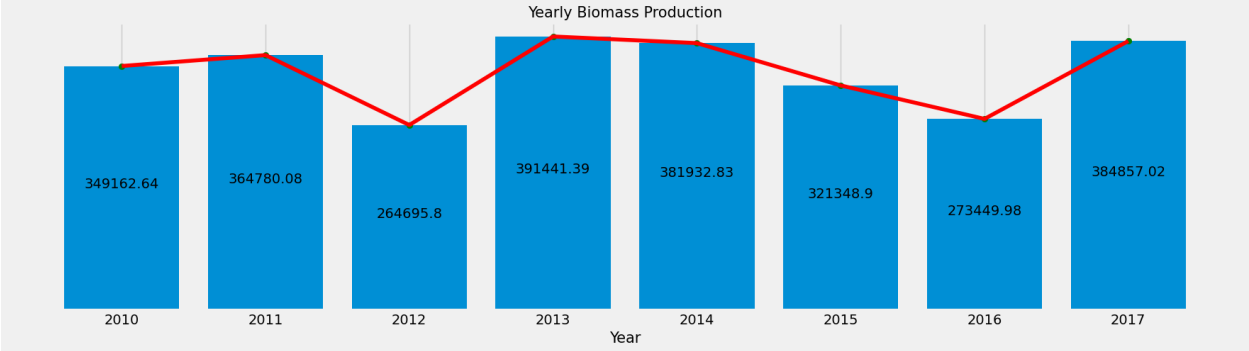
Problem Statement



The primary challenge entails designing an optimized spatial supply chain for biofuel production in Gujarat, Western India, wherein the yearly varying residual biomass distribution across the state's grid blocks serves as feedstock. The goal is to strategically locate Harvesting Sites based on historical biomass availability forecasts, efficiently collecting and transporting biomass bales to preprocessing Depots for densification and pelletisation. These processed pellets are subsequently distributed from Depots to fixed-location Biorefineries, aligning quantities with each Biorefinery's demand. The objective is to minimize transportation costs while meeting the biofuel production requirements and accommodating the changing spatial distribution of biomass resources.

Problem Breakdown

- **Harvesting Sites**
    - 2018 and 2019 biomass availability forecast for the harvesting locations.



- **Preprocessing Depots**
    - Determine the required number of depots by considering the total forecasted biomass availability and strategically position these depots.
    - Optimize the distribution of biomass from different harvesting locations to the designated depots.

- **Biorefinery**
    - Determine the required number of biorefineries based on the total pellets produced in the depots and strategically position these biorefineries.
    - Optimize the distribution of pellets from different depot locations to the designated biorefineries.

Solution Approach

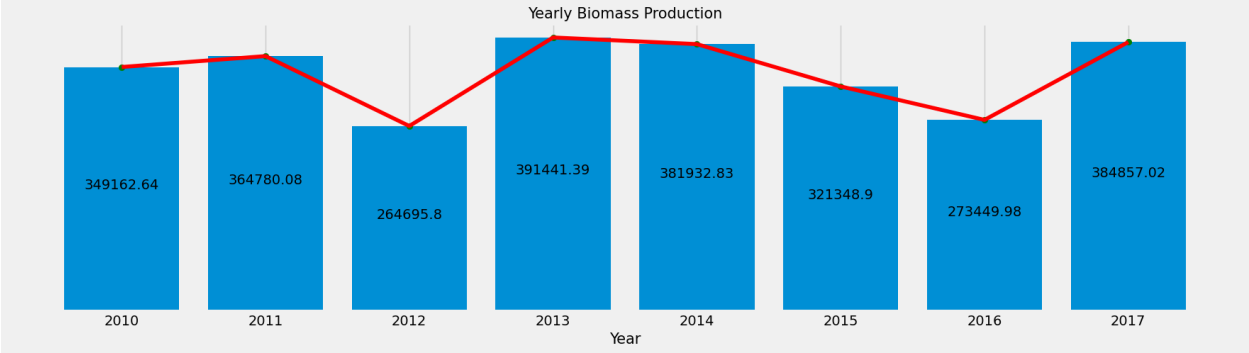
Approach 1: Biomass Forecasting

The dataset provided contains geospatial information (latitude and longitude) for the different grid points in the selected location, Gujarat, and historical biomass availability data spanning eight years (2010 to 2017).

```
DemandHistory.head()
```

	Latitude	Longitude	2010	2011	2012	2013	2014	2015	2016	2017
0	24.66818	71.33144	8.475744	8.868568	9.202181	6.023070	10.788374	6.647325	7.387925	5.180296
1	24.66818	71.41106	24.029778	28.551348	25.866415	21.634459	34.419411	27.361908	40.431847	42.126945
2	24.66818	71.49069	44.831635	66.111168	56.982258	53.003735	70.917908	42.517117	59.181629	73.203232
3	24.66818	71.57031	59.974419	80.821304	78.956543	63.160561	93.513924	70.203171	74.536720	101.067352
4	24.66818	71.64994	14.653370	19.327524	21.928144	17.899586	19.534035	19.165791	16.531315	26.086885

Data Analysis



The presented figure illustrates the annual biomass production across all harvesting sites. Notably, discernible trends emerge, with certain years (2012, 2010, and 2016) displaying lower biomass production, while other years (2011, 2013, 2014, and 2017) exhibit higher biomass production levels. To effectively incorporate this trend into the model, a data reconstruction process will be employed.

Data Reconstruction

In order to effectively capture the underlying trend present within the provided dataset and facilitate the training of a precise forecasting model, a reconstruction process was initiated. This entailed selecting the biomass history encompassing a span of three years preceding the target year, along with incorporating the corresponding geospatial information, as predictor variables. Consequently, the biomass value pertaining to the fourth year was employed as the target variable for this model. Specifically, the biomass values for the years 2010-2012 were utilized as predictors for forecasting the year 2013, while the biomass values from 2011-2013 were utilized for predicting 2014, and this pattern was continued for subsequent years.

	Latitude	Longitude	year1	year2	year3	Target
0	24.66818	71.33144	8.475744	8.868568	9.202181	6.023070
1	24.66818	71.41106	24.029778	28.551348	25.866415	21.634459
2	24.66818	71.49069	44.831635	66.111168	56.982258	53.003735
3	24.66818	71.57031	59.974419	80.821304	78.956543	63.160561
4	24.66818	71.64994	14.653370	19.327524	21.928144	17.899586

Feature Engineering

Additional features, including aggregate and interaction features, were derived from the historical data to enrich the information available for the forecasting process.

year_avg	year_std	year2_year1_change	year2_year1_diff	year3_year1_change	year3_year1_diff	year3_year2_change	year3_year2_diff
8.848831	0.363620	0.046347	0.392824	0.085708	0.726437	0.037617	0.333612
26.149180	2.274009	0.188165	4.521570	0.076432	1.836637	-0.094039	-2.684933
55.975020	10.675464	0.474654	21.279533	0.271028	12.150623	-0.138084	-9.128910
73.250755	11.535388	0.347596	20.846886	0.316504	18.982124	-0.023073	-1.864761
18.636346	3.686310	0.318982	4.674154	0.496457	7.274775	0.134555	2.600620

Model Selection

- Standard LGBM
- objective = tweedie

Approach 2: Depots Identification and Positioning

To commence, the initial step involves determining the total count of depots required to handle the entire available biomass for a specific year, such as 2017. This is accomplished by dividing the total biomass quantity available for the designated year (e.g., 2017) by 20,000, which represents the maximum annual processing capacity of an individual depot, as stipulated within the competition guidelines.

```
[16] print(f"Yearly available biomass 2017: {DemandHistory['2017'].sum()}")
     print(f"Yearly Depot Capacity: 20,000")
     print(f"Total Depots neede: {int(np.ceil(DemandHistory['2017'].sum()/20000))}")

     Yearly available biomass 2017: 384857.021076038
     Yearly Depot Capacity: 20,000
     Total Depots neede: 20
```

The entire Gujarat region is then segmented into four clusters. The number of depots required for each cluster is determined by dividing the total biomass demand within the cluster by the capacity of a single depot and rounding up the value to the nearest integer.

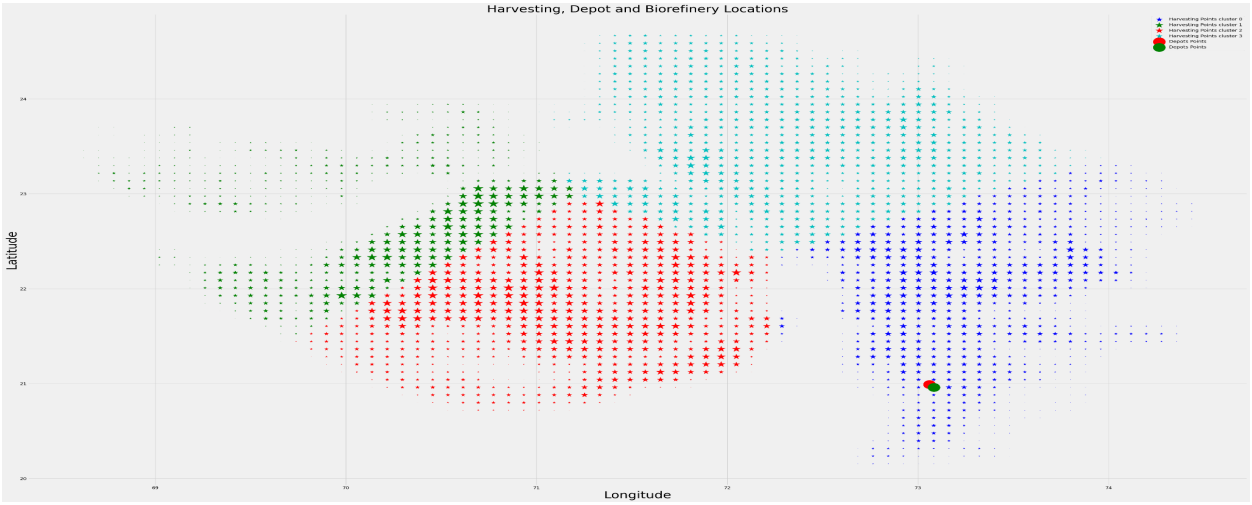
Clusters	2017
0	78317.907872
1	64589.835906
2	148786.716628
3	93162.560670

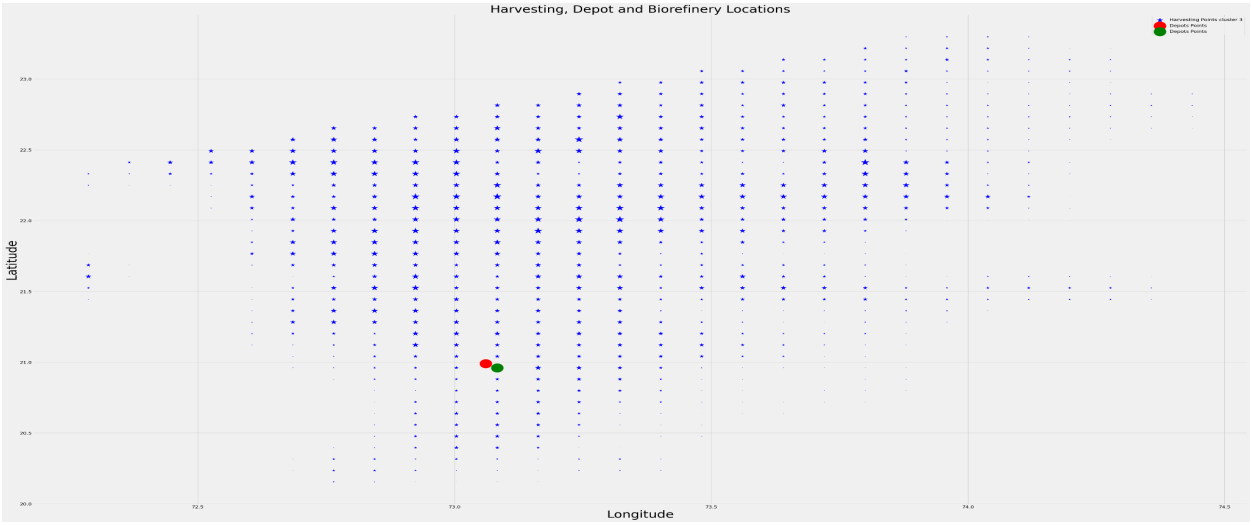
Clusters	2017
0	3.915895
1	3.229492
2	7.439336
3	4.658128

Clusters	2017
0	4.0
1	3.0
2	7.0
3	5.0

Each main cluster is subdivided based on the number of depots required (e.g., Cluster 0 in the image above requires 4 depots). These subdivisions create additional sub-clusters within the four main clusters, with each sub-cluster representing a potential location for a depot. To determine the depot's placement within each sub-cluster, the **center of gravity method** is employed.

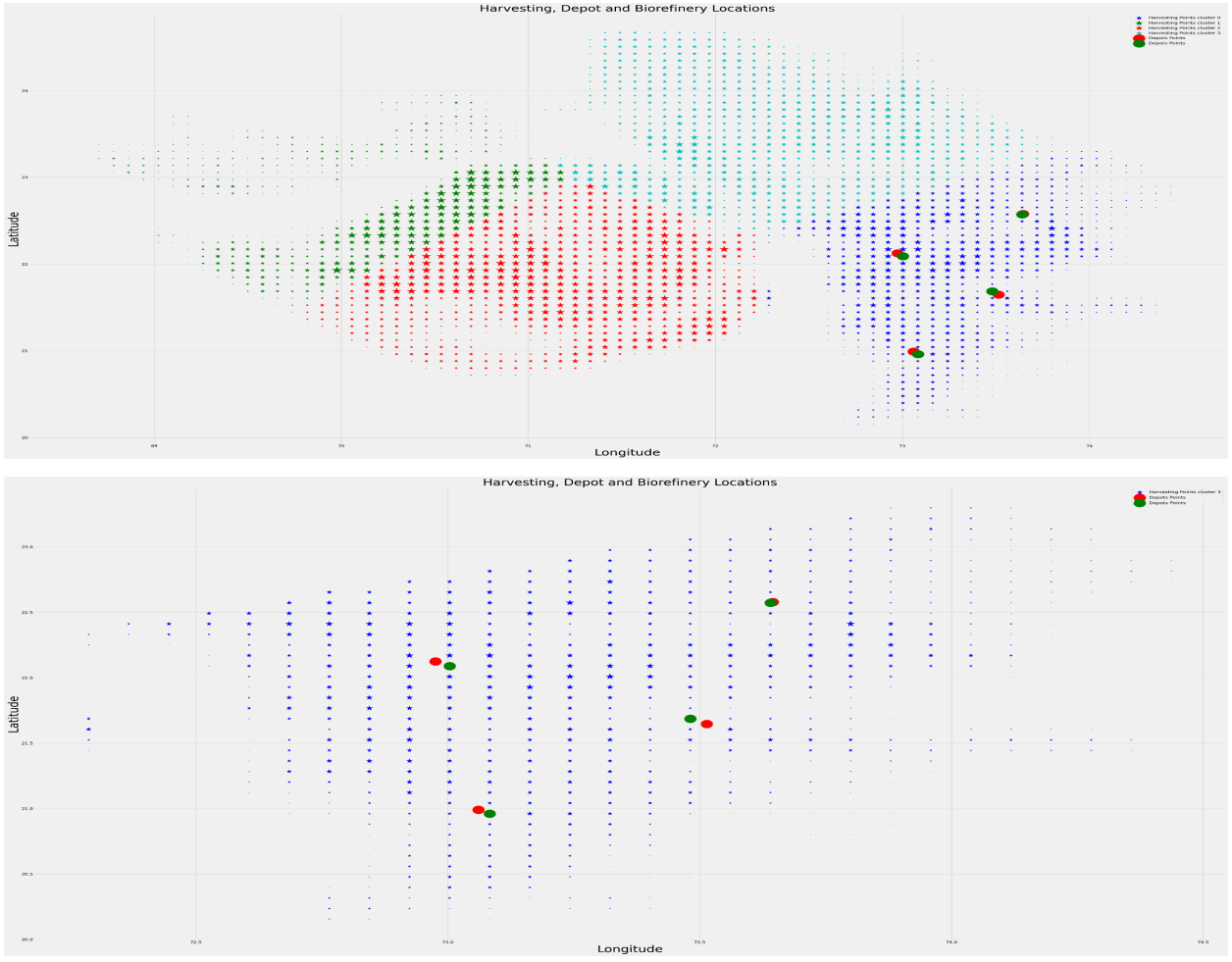
Center of gravity method: The center of gravity method is a technique used to find the best location for a facility within a given area. By calculating the weighted average of demand points' coordinates, it identifies the optimal point that minimizes transportation distance or cost.



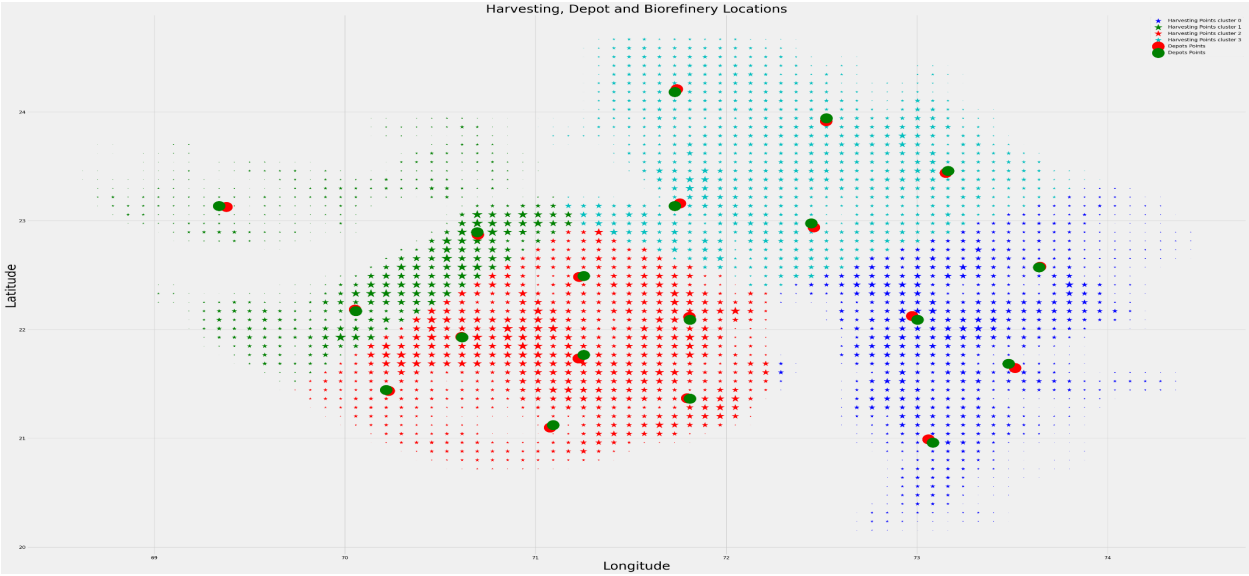


In the first image, there are four main clusters representing the entire region. The second image displays the first cluster, which contains two points, one red and one green. These points belong to a single sub-cluster within the first cluster. The red point indicates the optimal depot location determined using the center of gravity method.

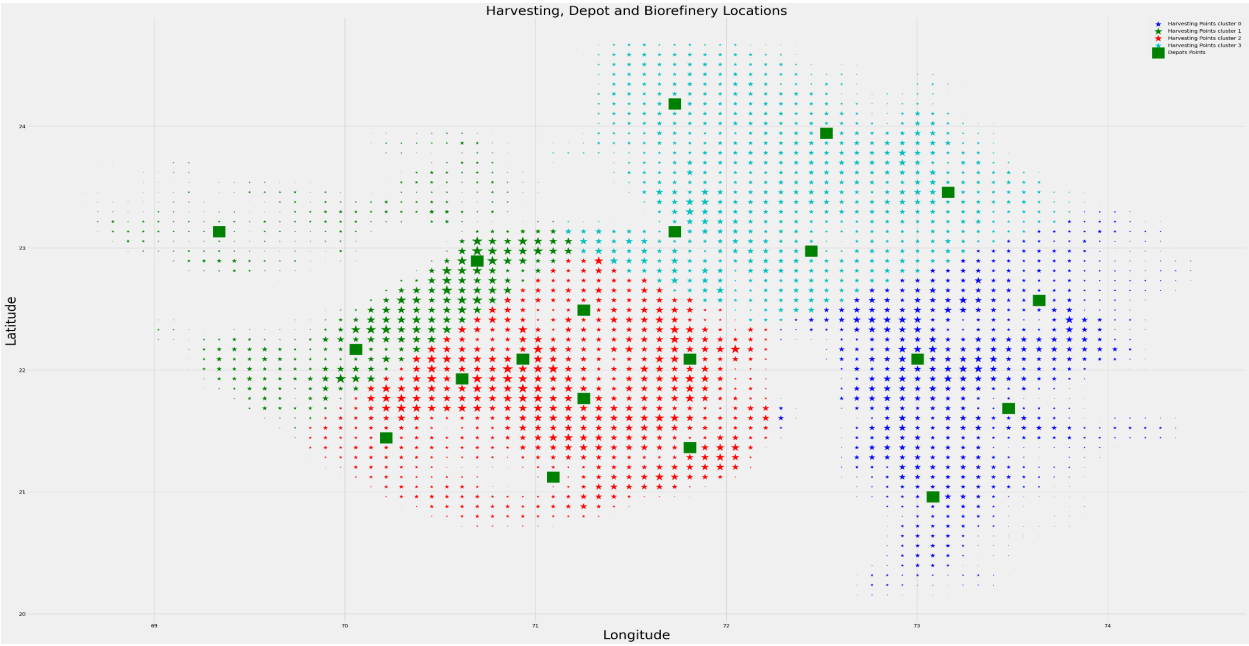
However, the competition requirement dictates that each depot must be located on a harvesting grid point, which the center of gravity function did not consider. To address this, the nearest grid location to the identified red point (represented by the green point) is determined using their **haversine distance**. This ensures that the depot is placed on the grid point with the shortest distance from the optimal center of gravity location.



The two figures above display the assigned depots for all the sub-clusters in the first cluster. The meaning of the red and green points remain the same as explained earlier.



The figures above display the assigned depots for all the sub-clusters in the four major clusters. The meanings of the red and green points remain the same as explained earlier.



The figures above display the assigned depots for all the sub-clusters in the whole region.

Approach 3: Biorefinery Identification and Positioning

To locate the biorefineries, the P-median approach was implemented. The P-median approach is a classic optimization technique used to optimally locate a specified number of facilities, such as biorefineries, in a way that minimizes the overall transportation costs from multiple depots. In this case, the goal is to optimally position multiple biorefineries to receive pellets from several depots while minimizing transportation expenses. Here's a breakdown of the process.

The model formulation is as follow:

Parameters:

- **N:** The set of potential refinery locations, indexed by $i = 1, 2, \ldots, n$
- **M:** The set of depot points, indexed by $j = 1, 2, \ldots, m.$
- d_{ij}: The distance (cost) between potential refineries*i* and demand point *j*.
- **p:** The number of facilities to be located.
- C_{ref}: Maximum amount of depots that can be served by 1 refinery = 5

Variables:

- x_{ij}: Binary variable indicating whether refinery *i* serves depot *j* (1 if it does, 0 otherwise).
- y_i: Binary variable indicating whether refinery *i* is open (1 if it is, 0 otherwise).

Objective: (Minimize the total cost)

- Minimize: $\sum_{i=1}^{n} \sum_{j=1}^{m} d_{ij} \cdot x_{ij}$

Constraints:

- Each depot point must be served by exactly one open refinery:

$$\sum_{j=1}^{m} x_{ij} = 1 \quad \text{for } i = 1, 2, \ldots, n$$

- Each refinery must serve less or exactly the maximum amount of depots it can serve:

$$\sum_{i=1}^{n} x_{ij} \leq C_{ref} \quad \text{for } j = 1, 2, \ldots, m$$

- The number of open facilities must be equal to **p**:

$$\sum_{i=1}^{n} y_i = p$$

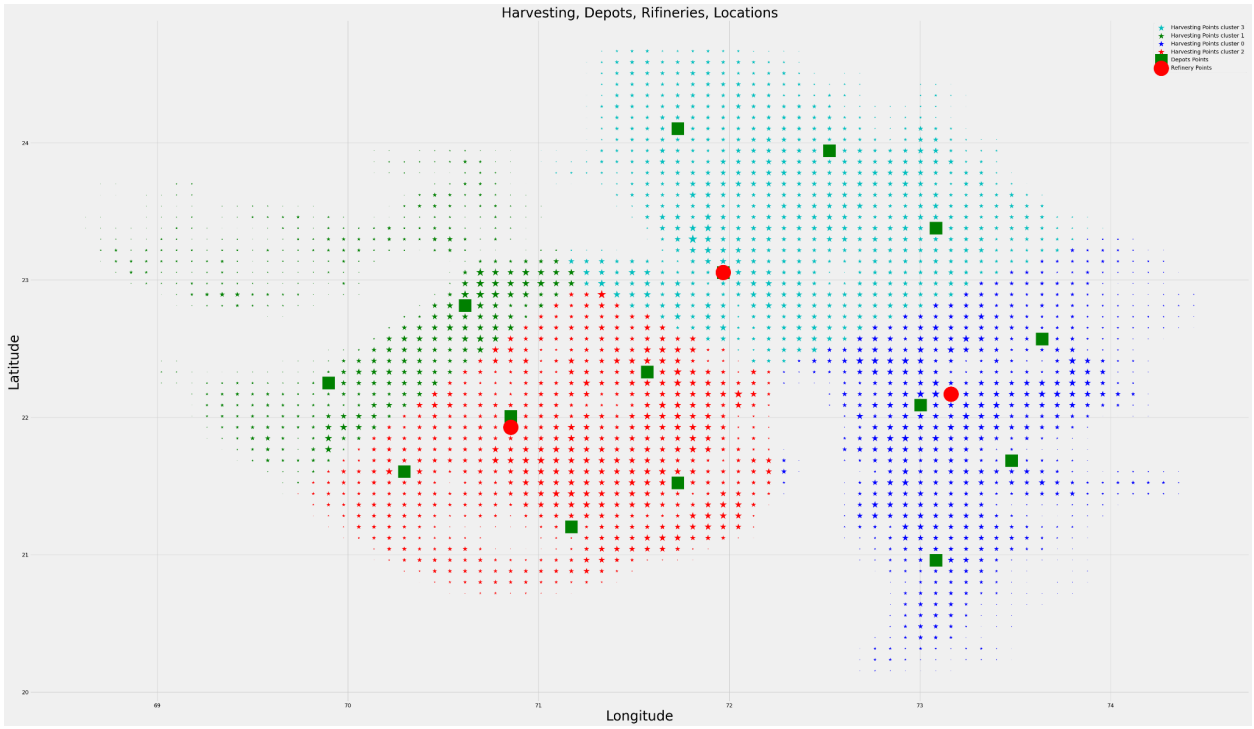
- If a facility is open, it must serve the associated demand points:

$$x_{ij} \leq y_i \quad \text{for } i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, m$$

- Binary constraints on variables:

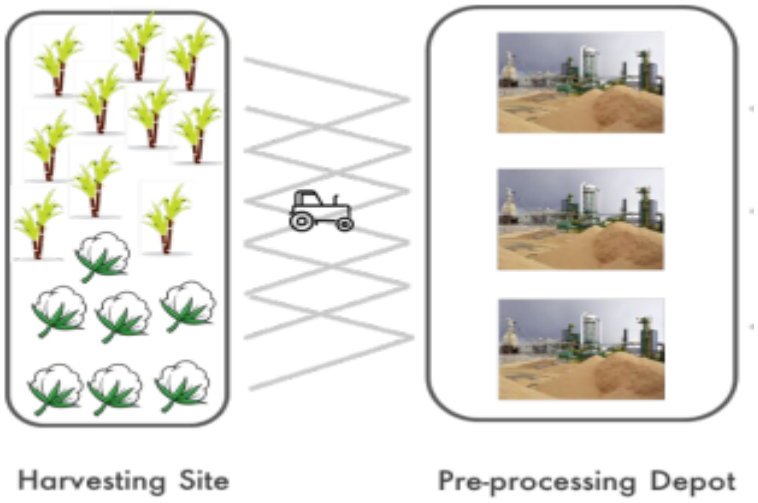
$$x_{ij} \in 0, 1 \quad \text{for } i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, m$$

$$y_i \in 0, 1 \quad \text{for } i = 1, 2, \ldots, n$$



The figure above shows the location of the biorefinery in red after implementing the results of the optimization determined by the P-median approach. These optimal positions ensure that pellets are distributed efficiently while minimizing transportation costs.

Approach 4: Optimal Biomass Distribution



To achieve the efficient distribution of harvested biomass from diverse harvesting sites to the designated depots, we employed the Transportation Model. This model is a powerful optimization technique designed to strategically allocate resources across different origins and destinations while minimizing costs and maximizing effectiveness.

Formulation:

Variables and Parameters:

- $Biomass_{ij}$: Biomass demand-supply matrix
- $Dist_{ij}$: Cost of transporting one unit of biomass from Harvesting Location *i* to Depot *j*.
- $Biomass_{forecast,i}$: Supply capacity of Harvesting site *i*.
- Cap_{depot}: Maximum yearly processing capacity of a depot (20,000).

Decision Variables:

Let $Biomass_{ij}$ represent the quantity of biomass transported from Harvesting Location *i* to Depot *j*.

Objective Function:

Minimize the total transportation cost:

$$\text{Minimize} \sum_i \sum_j Dist_{ij} \cdot Biomass_{ij}$$

Constraints:

Biomass Supply Constraints:

The total supply from each harvesting site should not exceed its capacity:

$$\sum_j Biomass_{ij} \leq Biomass_{forecast,i} \quad \text{for each harvesting point } i$$

The total supply from each harvesting site must exceed 80% of its capacity:

$$\sum_j Biomass_{ij} \geq 0.8 \cdot Biomass_{forecast,i} \quad \text{for each harvesting point } i$$

Biomass Demand Constraints:

The total biomass supplied to each depot should be less than or equal to its capacity:

$$\sum_i Biomass_{ij} \leq Cap_{depot} \quad \text{for each depot } j$$

Non-Negativity Constraints:

The quantities of biomass transported cannot be negative:

$$Biomass_{ij} \geq 0 \quad \text{for all } i \text{ and } j$$

Approach 5: Optimal Pellet Distribution

To achieve the efficient distribution of pellets from diverse depots to the designated biorefineries, we employed the Transportation Model.

Formulation:

Variables and Parameters:

- $Pellete_{ij}$: Pellete demand-supply matrix
- $Dist_{ij}$: Cost of transporting one unit of Pellete from Depot *i* to Refinery *j*.
- $Pellete_{cap,i}$: Supply capacity of Depot *i*.
- $Cap_{refinery}$: Maximum yearly processing capacity of a Refinery (100,000).

Decision Variables:

Let $Pellete_{ij}$ represent the quantity of Pellete transported from Depot *i* to Refinery *j*.

Objective Function:

Minimize the total transportation cost:

$$\text{Minimize} \sum_i \sum_j Dist_{ij} \cdot Pellete_{ij}$$

Constraints:

Pellete Supply Constraints:

The total supply from each depot should not exceed its capacity:

$$\sum_j Pellete_{ij} \leq Pellete_{cap,i} \quad \text{for each depot } i$$

The total supply from each depot must exceed 80% of its capacity:

$$\sum_j Pellete_{ij} \geq 0.8 \cdot Pellete_{cap,i} \quad \text{for each depot } i$$

Pellete Demand Constraints:

The total demand at each refinery should be less than or equal to its capacity:

$$\sum_i Pellete_{ij} \leq Cap_{refinery} \quad \text{for each refinery } j$$

Non-Negativity Constraints:

The quantities of Pellete transported cannot be negative:

$$Pellete_{ij} \geq 0 \quad \text{for all } i \text{ and } j$$

Project Structure

```
├── LICENSE
├── README.md
├── data
│   ├── raw                            <- Downloaded datasets
│   │   ├── Biomass_History.csv
│   │   ├── Distance_Matrix.csv
│   │   └── sample_sbmission.csv
│   └── output
│       └── Submission.csv
├── models                         <- Folder to contain the saved models
├── notebooks
│   ├── Analysis_notebook(ShellAI).ipynb              <- Analysis Notebook
│   ├── scripts_runner_notebook.ipynb         <- Notebook for running the script
│   └── complete_model_implementation.ipynb   <- Notebook for complete implementation of the project (From EDA to submission)
├── reports
│   └── figures                      <- Generated graphics and figures
├── requirements.txt                 <- Requirements text file
├── src                              <- Source code for use in this project.
│   ├── __init__.py                  <- Makes src a Python module
│   ├── config.py                    <- Configuration file
│   ├── utils.py                     <- Python script contatining the necessary utilities
│   ├── depot_locator.py             <- Configuration file
│   ├── optimization_model.py        <- Script for linear programming.
│   ├── refinery_locator.py             <- Script to mske prediction snd create the submission file.
│   ├── test_constraint.py           <- Script to confirm submission if not going against any of the constraint
│   └── visualize.py                 <- Script to generate the analysis graphics.
└── test_environment.py              <- Script to confirm the correct python environment.
```

Project Repository: [Click here]