Detailed Report: Efficient and Fair Line Construction Project

The Efficient and Fair Line Construction project aims to construct a gas pipeline in a straight line that serves a set of houses with coordinates given by latitude and longitude. The project focuses on optimizing the pipeline placement to achieve efficiency and fairness while considering the distances from houses to the pipeline.

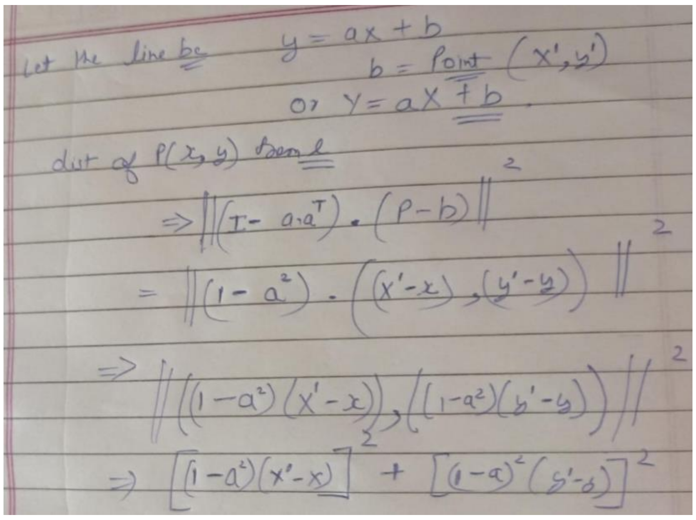
Objective 1 aims to minimize the total distance from each house to a line, defining efficiency as achieving a local minimum in this cost.

Objective 2 targets fairness, seeking to minimize the maximum distance from any house to the line, again aiming for a local minimum in this cost.

Objective 3 expands this to multiple lines, where efficiency is defined as minimizing the total distance from houses to any line in a set of k lines.

Challenges Encountered:

# One of the major challenges we encountered is in the deciphering the formulae which was provided in the Objective 1 for calculating the distance to solve <u>this we need to see some videos on Linear algebra and vectors</u> from YouTube, then we were able to crack it and here it is:



Let the line be $y = ax + b$

$b = \text{Point } (x', y')$

or $Y = aX + b$.

dist of $P(x, y)$ from l

$$\Rightarrow \| (I - a \cdot a^T) \cdot (P - b) \|^2$$

$$= \| (1 - a^2) \cdot ((x' - x), (y' - y)) \|^2$$

$$\Rightarrow \| ((1 - a^2)(x' - x)), ((1 - a^2)(y' - y)) \|^2$$

$$\Rightarrow [(1 - a^2)(x' - x)]^2 + [(1 - a)^2 (y' - y)]^2$$

# Next Challenge we encountered is in the visualizing of the Houses of California in the real map, then we found a library called Basemap from google, we read the documentation of it and implemented the feature.

Link of Basemap: https://matplotlib.org/basemap/stable/

# Third Challenge we encountered is in the figuring out which kind of problem is this, and we sooner figured out it is not any conventional Regression or Classification, but Similar to Geometric Optimization. Since we could not find the relevant Algorithm for and we have no prior Knowledge of it, We Firstly tried to Implement the Brute Force, Moving upto each iterations and it took around 12100 epochs for the first time around 1.5hr of exection and in second run we reduced it to 8100 epocs

Key Decisions Made

-**Algorithm Selection**

We first Applied **Brute Force**: It does not have the us with the accurate fit of the Gas Pipeline.

Then we moved to **Median Line Estimator**: It provided us with descent fit Line.

Followed by this, we moved to **Ransac Estimator and Gradient Descent** (to optimize the loss function).

-**Programming Languages**: Used Python for implementation, leveraging its rich libraries like scikit-learn.

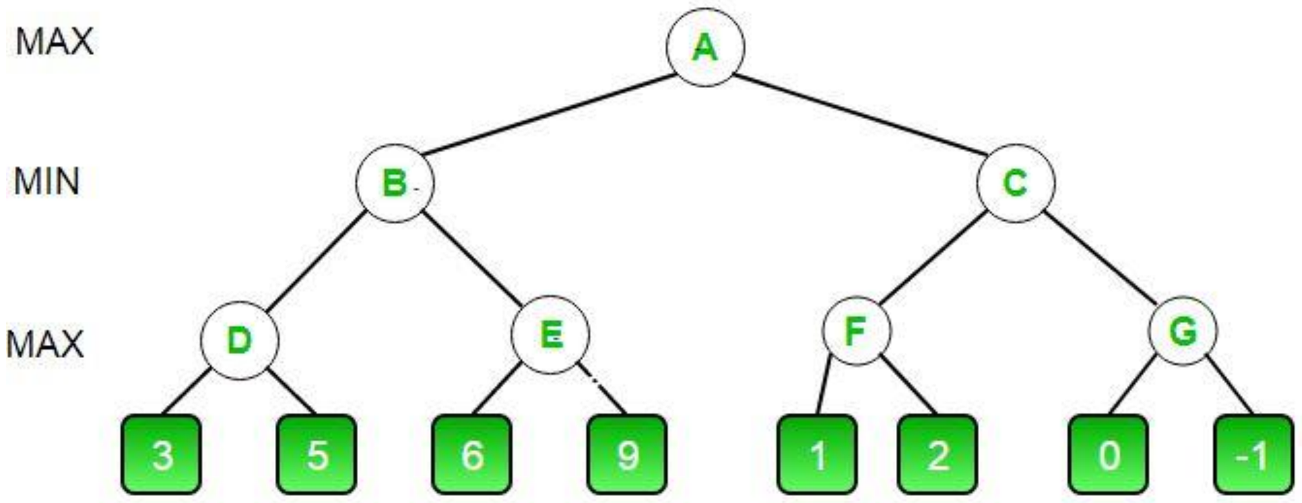
-Data Visualization Tools: Utilized 'matplotlib' , 'seaborn' and 'Basemap' for visualizing geographical data and pipeline placement.

Solutions Implemented

Algorithmic Solutions

#Brute Force: Implemented the Brute Force algorithm to optimize pipeline placement for efficiency and fairness.

#Minmax For Objective 2:



#Fairness Criteria Implementation: Integrated fairness criteria into the pipeline placement algorithm.

#Optimization Techniques: Used optimization algorithm (Gradient Descent) to enhance the efficiency and fairness of the pipeline placement process.

Results and Performance Evaluation

#Qualitative results of the different algorithm

Evaluation of the algorithm's performance in terms of efficiency: Median Line Estimator and Ransac Estimator gives us the satisfying results out of our implemented methods.

Conclusion

The project successfully addressed the challenges of efficient and fair line construction by designing algorithms that optimize pipeline placement while considering location constraints The implemented solutions demonstrated effective pipeline placement strategies and achieved significant improvements in efficiency and fairness.