Approach

We used Google's Geocoding API to geocode addresses into latitude and longitude, which was then used to find the nearest road using Google's Road API. The distance between the road and the storefront was calculated using the haversine formula, converting it into the bearing between the two points. Using the coordinate of the storefront, we used Google's Street Vision API to obtain an image which is temporarily saved. We then created a json key for using Gemini's cloud vision API to quantify the obstruction in the saved Street View image. The Overpass API is used to convert the address entered into an OSM index. The data set is searched for a matching OSM index and the variables in the matching row are calculated using the formula described below.

Assumptions:
- Factors affecting visibility are similar in all states, using Georgia as an example for training.
- The street view location represents a single, straight road segment, that is, no roads with multiple observation points.
- Adjusting the heading for the street view API based on bearing is enough to ensure a
- A smooth terrain between the storefront and the viewpoint, for bearing calculation.
- Though the attenuation coefficient realistically decreases with increase in distance from storefront, we assumed a coefficient of 0.1 throughout, since the distances in the dataset would result in a negligible change in the coefficient.

Data Preprocessing Steps

We cleaned the data, dropping the following:
- We used the parquet file because it had the fastest performance.
- 'country_code', 'state-code', 'county_fips' - since working the dataset was very resource intensive, we decided to build the demo based on Georgia data only. The county variable is disregarded as well because the API fetches the coordinates, which is the variable we use to obtain traffic volume.
- 'vmt', 'year', 'date' since all data is from March 2023.
- rows having different indexes but same values for all other variables being used in calculations, were merged.

Evaluation Metrics

D - Distance from the road
θ - Angle of view
O- Obstructing objects

Mathematical/statistical Reasoning

$V = f(D, \theta, O)$

- Since visibility decreases with distance, we use an exponential decay function:
  $V_D = e^{-\alpha DV}$

Where α is an attenuation coefficient.

- $V_\theta = \cos^2(\theta)$

The cosine of the angle between the observer's view direction and the normal to the storefront is taken and squared to account for the significant differences with change in angle.

- $V_O = 1 - O$

The obstruction is subtracted from one to calculate the percentage of storefront that is not obstructed.

- $V = V_D \cdot V_\theta \cdot V_O$

$$V = e^{-\alpha D} \cdot \cos^2(\theta) \cdot (1 - O)$$

The visibility is the product of the three values, multiplied by the traffic volume, which is calculated as follows:

$$\text{Scaling Factor} \approx \frac{\sum \text{trips\_volume}}{\sum \text{trips\_sample\_count}}$$

$$\text{Estimated Traffic Volume} = \frac{\text{trips\_volume}}{\text{trips\_sample\_count}} \times \text{Scaling Factor}$$

We normalize the visibility score into a feasible range (0 to 1) and then display it as a percentage.

Potential real-world applications

It's useful as a metric to improve store visibility. It might be especially useful for small businesses. Using generative AI to identify movable objects can serve as strong recommendations for realtors. For unmovable objects, other factors could be changed.

Limitations

- Nearest road might differ according to format of address, since Google Maps has multiple addresses for one store in some cases.
- The solution is dependent on the accuracy of the Google Street View API, which might sometimes result in inaccurate results, especially in terms of obstruction.
- Stores located at corners could have multiple storefronts. The algorithm only considers one of these storefronts, depending on minimum differences in distance.

Scalability of the Solution

- Since the solution does not involve training a model, it would be simple to use a larger dataset with more computing power. The algorithm relying on few variables minimizes this.
- The accuracy of the scaling factor and thus the traffic volume is highly dependent on how thoroughly representative the sample data is of the population traffic. Accuracy can be increased by using a machine learning model. A predictive model could also incorporate from external sources other factors that might affect traffic volume, like speed limits and weather conditions.