Route Optimization Tool Documentation

By Stella Wong

Background:

1. **Customer Service Delays:**
   As the economy recovers from the COVID-19 pandemic, many customers are calling to resume waste collection services or initiate new business with us. However, our customer service team is unable to confirm service days immediately. They must wait for the dispatch team to assign the customer to a route before calling back, which delays service confirmation and affects customer satisfaction.

2. **Rising Transportation Costs:**

   Financial reviews have shown that transportation expenses are rising, and we are also experiencing a driver shortage. While we are not modifying existing customer route schedules, there is an opportunity to optimize routing for new or returning customers to reduce operational costs.

Objective:

To streamline operations and enhance customer service quality, this tool was developed to assist customer service representatives in assigning potential customer locations to the most appropriate route. More importantly, it seeks to minimize transportation costs by identifying the most efficient existing route.

Modeling:

For each existing route, the model performs the following steps:

1. **Calculates the distance** between the potential customer location and each stop on the route.
2. **Identifies the shortest distance** from the potential location to any stop on that route.
3. **Compares these shortest distances** across all existing routes.
4. **Selects the route** with the overall minimum distance to the potential location, under the assumption that it will require the least additional travel time to integrate the new stop.

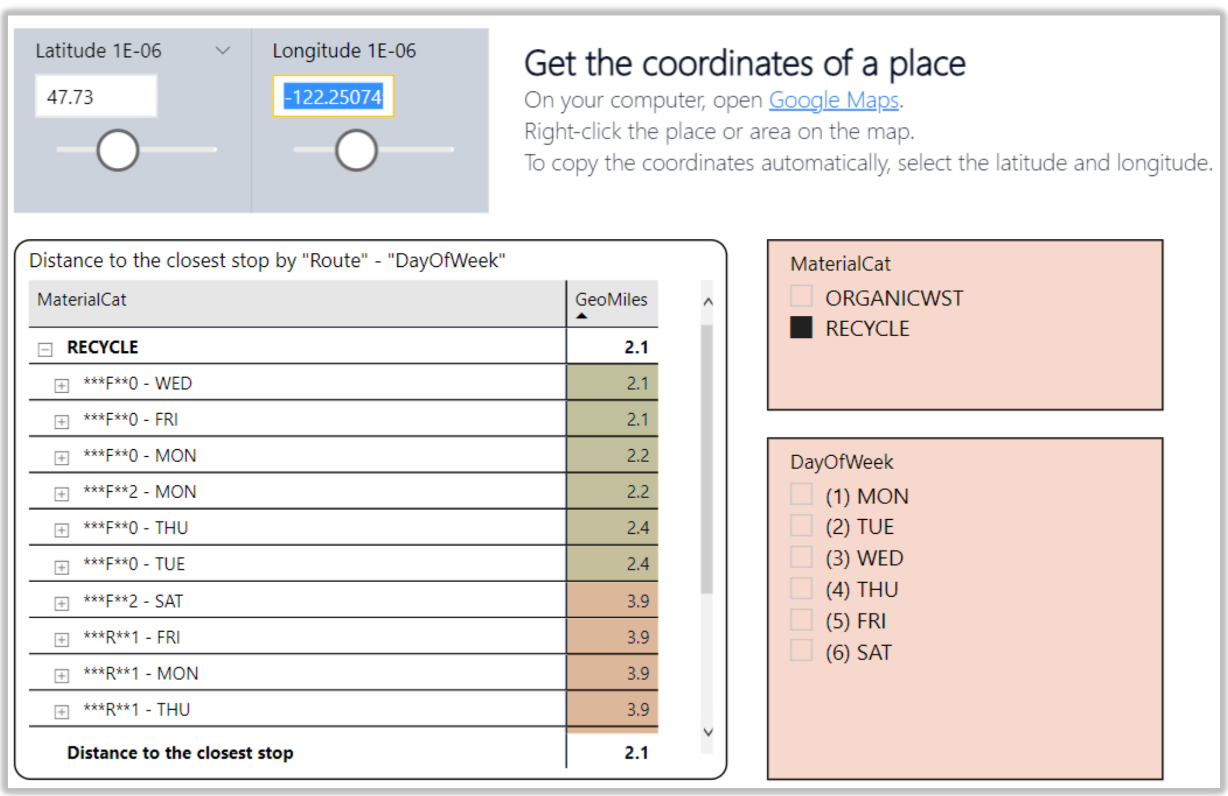
This approach ensures that new or returning customer locations are added to the most geographically efficient route, helping reduce fuel usage, time on the road, and overall trucking costs.

Instructions:

1. Obtain the geocode of the potential location from [Google Maps](), then enter the latitude and longitude values separately into their respective fields.
2. Select the material category of waste requested by the customer, such as *organic waste* or *recycle*.
3. If the customer has specified a preferred day of the week, apply the corresponding filter. If no routes are available for that day, the distance table will return no results.
4. The distance table displays the distance (in geo miles) from the entered geocode to the closest stop on each existing route. The table is sorted by the “Route” – “DayOfWeek” combination, from shortest distance (top) to longest (bottom).
5. Use the +/- button to expand a “Route” – “DayOfWeek” combination and view all stops on that route, sorted with the closest stops at the top.

Example:

For a recycle customer with geocode 47.729271, -122.250749, the optimal scheduling options are *Route ***F**0* on Wednesday (Top 1) or *Route ***F**0* on Friday (Top 2). Both routes have a closest existing stop only 2.1 geo miles away from the customer’s location.



Note: Parts of the route number have been replaced with asterisks (*) for privacy reasons.

DAX Code:

1. Use a SQL query to generate existing route information — route, day of week, latitude and longitude of stops, material category, etc. — and load the table into Power BI, naming **ExistingStops**.
2. Create measures to capture the geocode of an existing stop:

```
stop_x = IFERROR(VALUES('ExistingStops'[Latitude]), BLANK())

stop_y = IFERROR(VALUES('ExistingStops'[Longitude]), BLANK())
```

3. Create parameter tables for potential latitudes and longitudes in Power BI using GENERATESERIES:

```
Latitude = GENERATESERIES(47, 49, 1E-06)

Longitude = GENERATESERIES(-123, -121, 1E-06)
```

4. Create measures to capture the geocode entered in the tool:

```
Latitude Value = SELECTEDVALUE('Latitude'[Latitude])

Longitude Value = SELECTEDVALUE('Longitude'[Longitude])
```

5. Apply the **Haversine formula** to find the distance (in geo miles) between the entered geocode and stops in an existing route:

```
GeoMiles =

VAR r = 3958.7558657441

VAR dx = [Latitude Value]-[stop_x]

VAR dy = [Longitude Value]-[stop_y]

VAR a = POWER(SIN((dx*PI()/180)/2), 2) + (COS([Latitude
Value]*PI()/180)*COS([stop_x]*PI()/180)*POWER(SIN((dy*PI()/180)/2), 2))

VAR c = 2*ATAN(SQRT(a)/SQRT(1-a))

VAR d = r*c

RETURN d
```

6. Use MINX to compute the closest distance:

```
MinGeoMiles = MINX ('ExistingStops', [GeoMiles])
```

Note: The measure MinGeoMiles has been renamed as "GeoMiles" on the tool (see screenshot).

Reference:

1. Project 3, Part B: Descriptive Spatial Statistics | GEOG 586: Geographic Information Analysis (psu.edu)
2. SPATIAL STATISTICS (arcgis.com)