Snappfood | Commercial

Data Analyst Technical Task

Report of Task 1 - Restaurant Assignment

Prepared By: **Amirreza Akbari**

Objectives

The objective of this task was to design a method for assigning 500 restaurants to 5 field data collection agents in such a way that minimizes the overall travel cost. The only available features were two spatial coordinates: `feature_1` and `feature_2`, which represent the relative positions of each restaurant. The final dataset was expected to include a `person` column indicating the assigned agent for each location (from 1 to 5).

Methodology

The problem was framed as a spatial clustering challenge, aiming to segment the dataset into five geographically compact and balanced regions. Two unsupervised learning methods were implemented using `scikit-learn`:

- KMeans Clustering with `n_clusters=5`
- Agglomerative Clustering using Ward linkage and `n_clusters=5`

Each method provided cluster labels which were mapped to agents (`person = cluster_label + 1`). Visualizations were created to show how the restaurants were spatially divided among the agents.

In addition to generating the clusters, I incorporated travel cost estimation using two key metrics:

1. **Sum of Euclidean Distances to Cluster Centroid**

   Reflects spatial compactness and how centralized the restaurants are for each person.

2. **Estimated TSP (Traveling Salesman Problem) Route Distance**

   For each cluster, I calculated the shortest route that visits all restaurants in that cluster using a greedy TSP approximation via `networkx`. This is a more realistic proxy for actual field travel effort than centroid distances.

Evaluation

Each clustering method was evaluated based on:

- Silhouette Score: How well-separated and dense the clusters are
- Total Sum of Centroid Distances: Compactness of clusters
- Total TSP Route Distance: Approximated travel cost
- Standard Deviation of Cluster Sizes: Balance of workload (number of restaurants)
- Standard Deviation of TSP Distances: Fairness in travel burden among agents

Metric	KMeans	Agglomerative
Silhouette Score	0.597	0.585
Total Centroid Distance	181.76	184.34
Total TSP Distance	55.89	55.87
Cluster Size Std. Dev	0.84	1.28
TSP Distance Std. Dev	1.58	7.00

Final Selection

After evaluating both clustering methods based on multiple performance metrics, KMeans was selected as the preferred approach. While Agglomerative Clustering achieved a slightly lower total TSP distance (55.87 vs. 55.89), the difference is negligible and within a margin of error for heuristic-based route approximations.

KMeans demonstrated better results in the following key aspects:

- **Higher Silhouette Score** (0.597 vs. 0.585), indicating more distinct and well-separated clusters.
- **Lower Cluster Size Standard Deviation** (0.84 vs. 1.28), meaning a more balanced distribution of workload across agents.
- **Much Lower TSP Distance Standard Deviation** (1.58 vs. 7.00), signifying more equal travel effort among the 5 agents — a crucial fairness factor in field operations.

Although Agglomerative Clustering was slightly more compact in TSP path length overall, KMeans provided superior balance, interpretability, and fairness, making it the more practical choice for the intended use case.

Therefore, **KMeans** was selected for generating the final labeled dataset and accompanying visualizations.

Final Outputs

- `task1.ipynb`

  The complete Jupyter notebook containing visualizations, clustering methods, and evaluation process

- `task1_final_assignment.csv`

  Contains the cleaned and labeled dataset

Additional Notes

- The TSP approximation was performed using `networkx`'s `traveling_salesman_problem()` with a greedy heuristic, and is based on Euclidean distances, not real-world road networks.
- This solution assumes all agents start from anywhere in their cluster and do not need to return to a central hub.
- The dataset was treated purely geometrically. In a real deployment, I recommend:
  - Incorporating real-world GPS coordinates
  - Using road network routing APIs (like OSRM or Google Maps)
  - Factoring in additional logistics such as traffic, agent starting points, or time windows

Conclusion

This solution demonstrates the application of unsupervised learning and path estimation techniques to a logistics-style optimization problem. KMeans clustering, paired with centroid and TSP-based evaluations, enabled the creation of a scalable and explainable method for dividing restaurant territories among multiple agents — providing both clarity in execution and practical utility for real-world field operations.