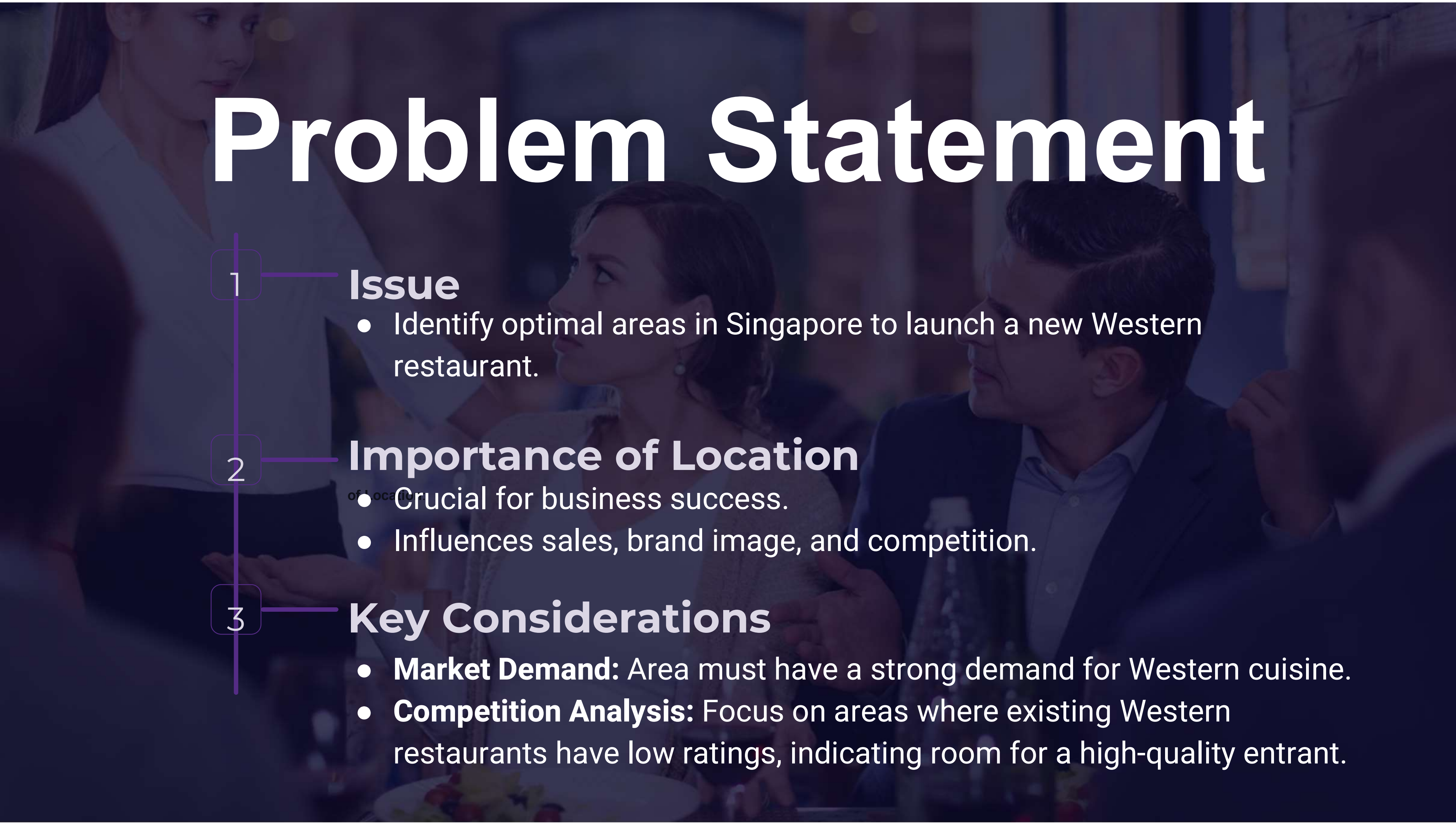
Western Specialty Restaurant Location Recommendation In Singapore (v2)

Prepared By

- Ong Weng Kai
- Ryan Kho Yuen Thian
- Thong Cheng How
- Yong Zee Lin

Assignment: Part 1

1.0 Introduction



Problem Statement

1 Issue

- Identify optimal areas in Singapore to launch a new Western restaurant.

2 Importance of Location

- Crucial for business success.
- Influences sales, brand image, and competition.

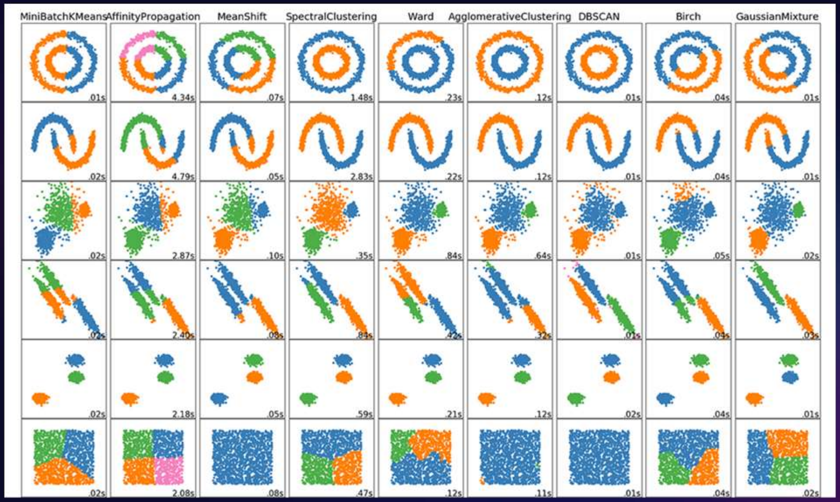
3 Key Considerations

- **Market Demand:** Area must have a strong demand for Western cuisine.
- **Competition Analysis:** Focus on areas where existing Western restaurants have low ratings, indicating room for a high-quality entrant.

Solution

1. **Kaggle Dataset**
   Grab Restaurants in Singapore (Unlabelled)

1. **Clustering Approach**
   - K-Means
   - BIRCH
   - DBScan
   - Agglomerative Hierarchical Clustering
   - Affinity Propagation



1. **Data Cleaning & Preprocessing**
   - Clean & Preprocess data for accuracy and consistency.
   - Filter to only include Western cuisine restaurants

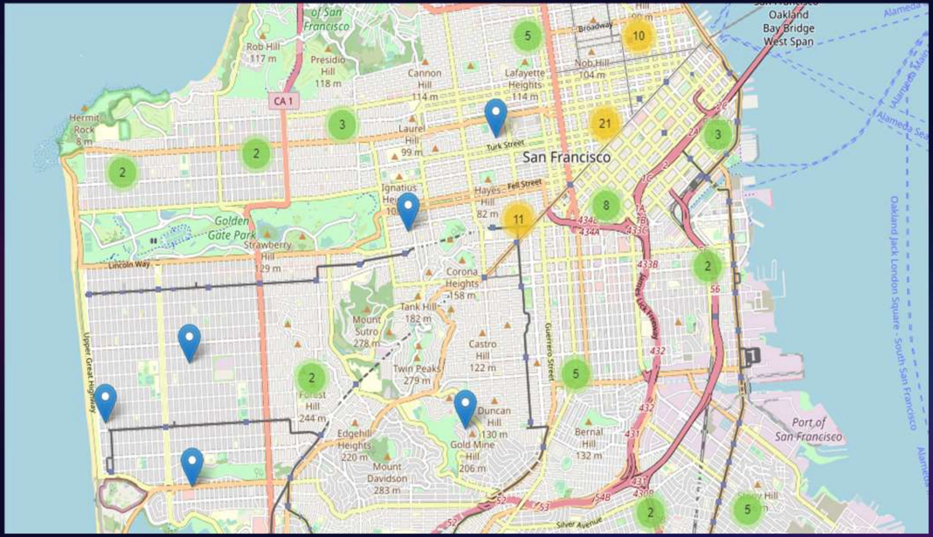
Solution

4. Analysis and Visualization

- Apply clustering algorithms to identify distinct groups.
- Analyse clusters using summary statistics & visualizations

5. Location Selection

- Pinpoint locations with high demand for Western food but lower-rated existing restaurants.
- Recommend optimal location(s) for the client's new Western restaurant



Objectives

1. To clean and preprocess the Grab Restaurants dataset for data integrity and consistency.

2. To perform EDA on the cleaned dataset to understand the data.

3. To apply 5 clustering techniques on the data.

4. To identify and visualise the distinct restaurant clusters formed.

Objectives

5. To assess the performance of 5 clustering techniques by using appropriate clustering performance metrics.

6. To display the locations of the clusters on the Singapore map

7. To interpret the clusters and recommend the best location(s) for the client.

2.0 Literature Review

5 Algorithms Covered

K-Means

BIRCH

Affinity Propagation

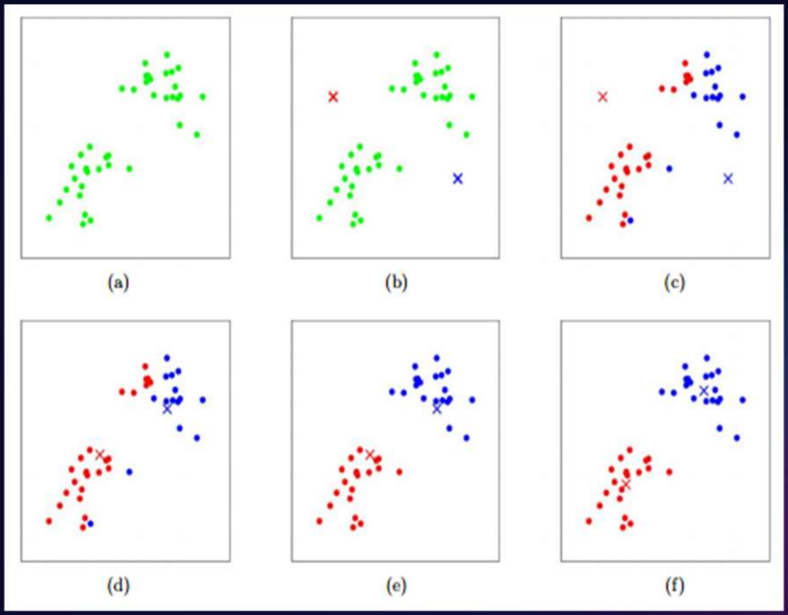
Agglomerative Hierarchical Clustering

DBSCAN

Algorithm 1: K-Means

Algorithm 1: K-means

- Partitioning-based Clustering Technique
- Allocates data points to one of the K clusters based on their proximity to the cluster centers
- Algorithm:
  a. Randomly select k items from a set of items
  b. Assign each item to the cluster it is most similar to
  c. Update the cluster centroids by computing the average value of the items within cluster
  d. Repeat steps b-c until no change in cluster.



K-Means

Application 1

Using K-Means for Recommending Restaurant Location

K-Means Application 1 (Kumar Shaswat, 2020)

- **Goal:**
  - To identify ideal localities for launching an Indian eatery in Delhi
- Ideal neighborhoods:
  - a scarcity of Indian restaurants
  - strong demand for them
  - potential for future expansion.
- Clustered the neighbourhoods into 5 clusters
  - Did not justify why k = 5
  - Did not explore other clustering algorithms

K-Means

Application 2

Topic Modelling and Clustering Restaurant Areas in Vancouver

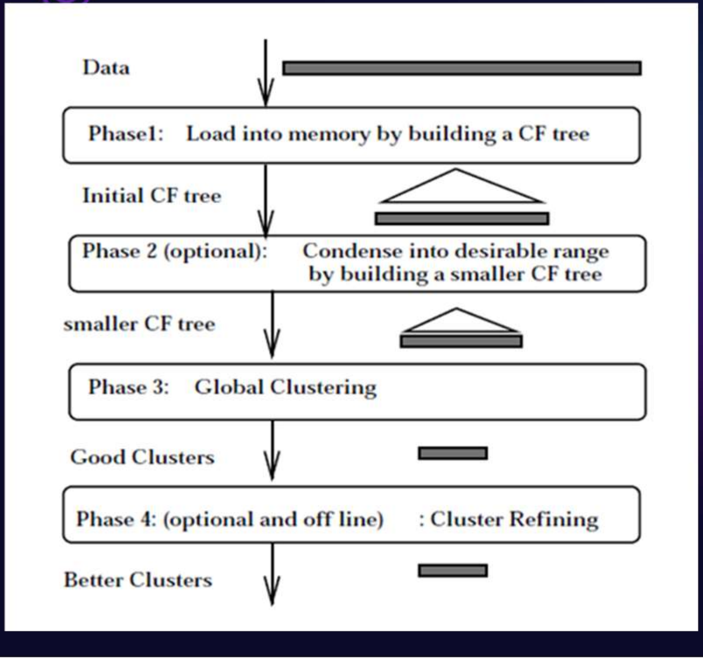
K-Means Application 2 (Lee, 2021)

- **Goal:**
  - To investigate the geographical distribution of particular cuisines and whether discernible patterns exist in restaurant clustering
- Travellers (Dining locales), Restaurateurs (Potential entry points)
- K-Means 2 Usages:
  - Cluster on the restaurant's location (using Silhouette Score)
  - Group neighborhoods based on their cuisine group weights (after Topic Modelling)

Algorithm 2: BIRCH

Algorithm 2: BIRCH

- Aka “Balanced Iterative Reducing and Clustering using Hierarchies”
- Handles large datasets
  a. Creates a condensed summary of the dataset
  b. Clusters the summary.
- 4 Phases:
  a. Loading
  b. Optional Condensing
  c. Global Clustering
  d. Optional Refining


Data
Phase1: Load into memory by building a CF tree
Initial CF tree
Phase 2 (optional): Condense into desirable range by building a smaller CF tree
smaller CF tree
Phase 3: Global Clustering
Good Clusters
Phase 4: (optional and off line) : Cluster Refining
Better Clusters

BIRCH

Application 1

A segmentation analysis mapping tool for the energy sector

BIRCH Application 1 (Liu et al., 2022)

- **Goal:**
  - To apply customer segmentation to analyse the daily energy consumption of residential buildings at the individual and at the neighbourhood level
- Resulted in SEGSys: An intuitive decision support tool for monitoring energy demand
- Some Reasons for Selecting BIRCH:
  - Identify anomalies
  - High efficiency
  - Excels with big data
  - Doesn't need the no. of clusters as input

BIRCH

Application 2

Recommendation for Location of Digital Signage using Fusion of Multiple Information Sources

BIRCH Application 2 (Xie et al., 2018)

- **Goal:**
  - To develop a sustainable model for suggesting location for digital signage
- **BIRCH**
  - One of the algorithms to split the study area into regions
  - 2nd highest Calinski-Harabasz Index
  - ↑ recommendation score threshold ↑ recommendation quality
  - Did not significantly affect recommendation results when compared to SOM

Algorithm 3:
Affinity Propagation

Algorithm 3: Affinity Propagation

Introduction

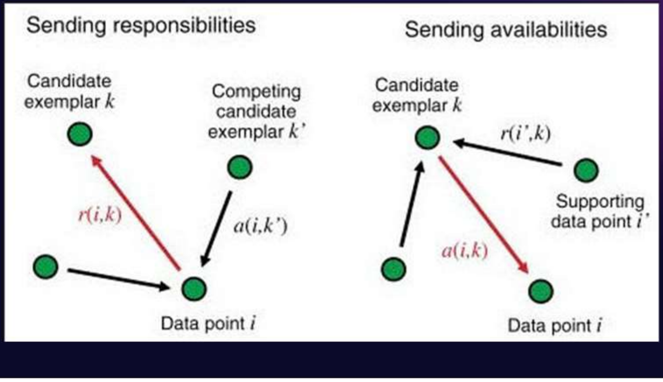
- Clusters are automatically identified without knowing the data point location.
- Utilizes message passing for each data point

Matrices

A. Similarity Matrix (S)
B. Responsibility Matrix (R)
C. Availability Matrix (A)

Key Steps

1. Similarity Calculation
2. Responsibility Calculation
3. Availability Calculation
4. Iterative Update
5. Net Responsibility Calculation
6. Exemplar Selection
7. Cluster Assignment



Affinity Propagation (AP)

Application 1

Food recommendation system using machine learning for diabetic patients

Affinity Propagation (AP) Application 1

(Phanich M et al., 2021)

Goal:

- Develop a food recommendation system for diabetic patients
- Label foods into three groups: normal, limited, and avoidable

How AP was Involved:

- 1 of the clustering algorithms
- Compared with K-Means, AP was slower but yielded more clusters
- Compared with SOM, AP took longer time and yielded less clusters

Affinity Propagation (AP)

Application 2

Multi-stage hierarchical food classification

Affinity Propagation (AP) Application 2

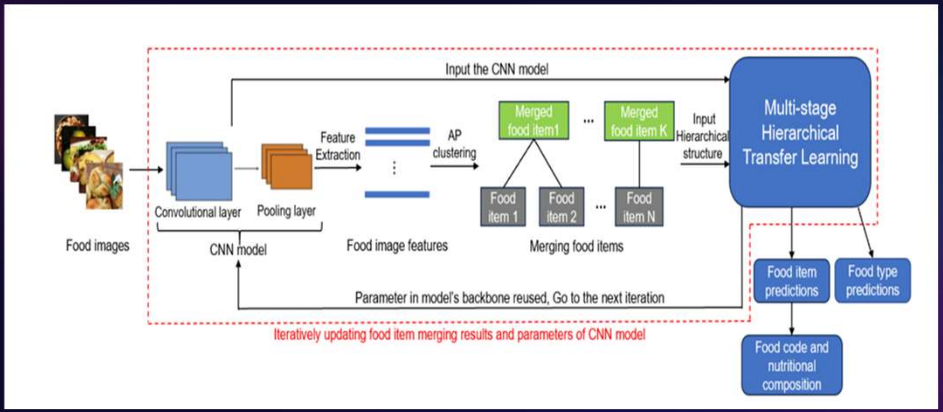
(Pan et al., 2023)

Goal:

- To classify food images with specific food ingredients containing nutrient information

How AP was involved:

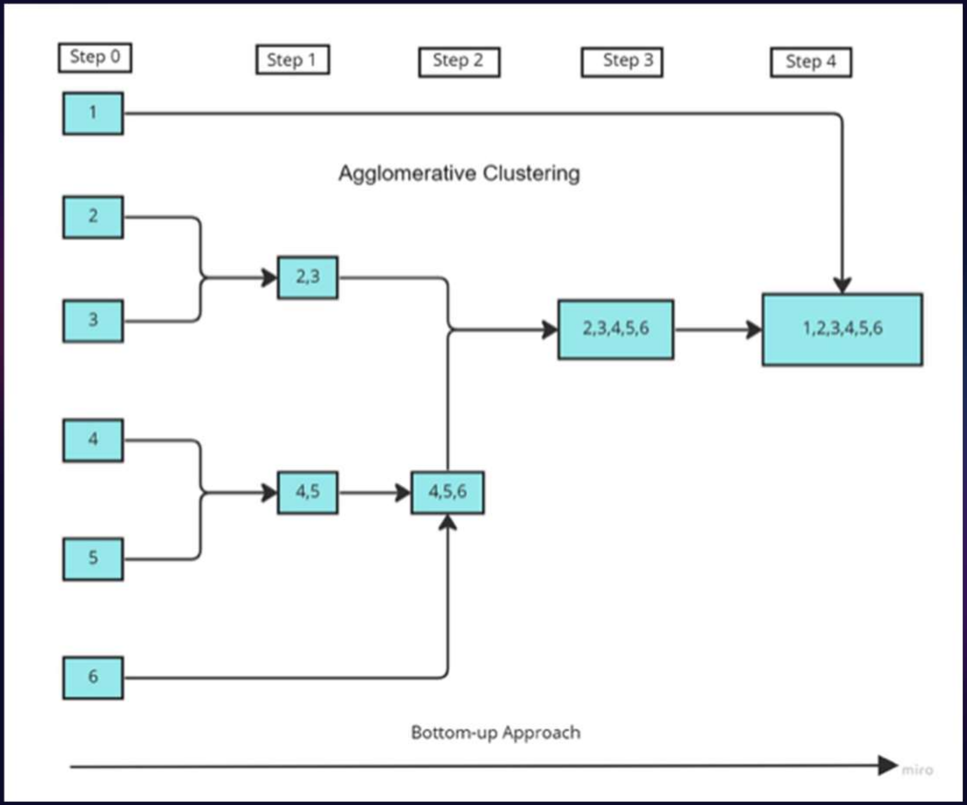
- Used for clustering and merging food images during the training process
- Flexibility in cluster identification without preset numbers.



Algorithm 4:
Agglomerative Hierarchical Clustering

Agglomerative Hierarchical Clustering

- Bottom-up approach: starts with individual cluster
- Calculates based on similarity score
- Process involves several key steps
  - Initialization
  - Proximity Matrix computation
  - Linkage, Merge
  - Update
  - Repeat
- useful for grouping data points



Agglomerative Hierarchical Clustering (AHC)

Application 1

Food ingredient Classification According to Nutritional Composition

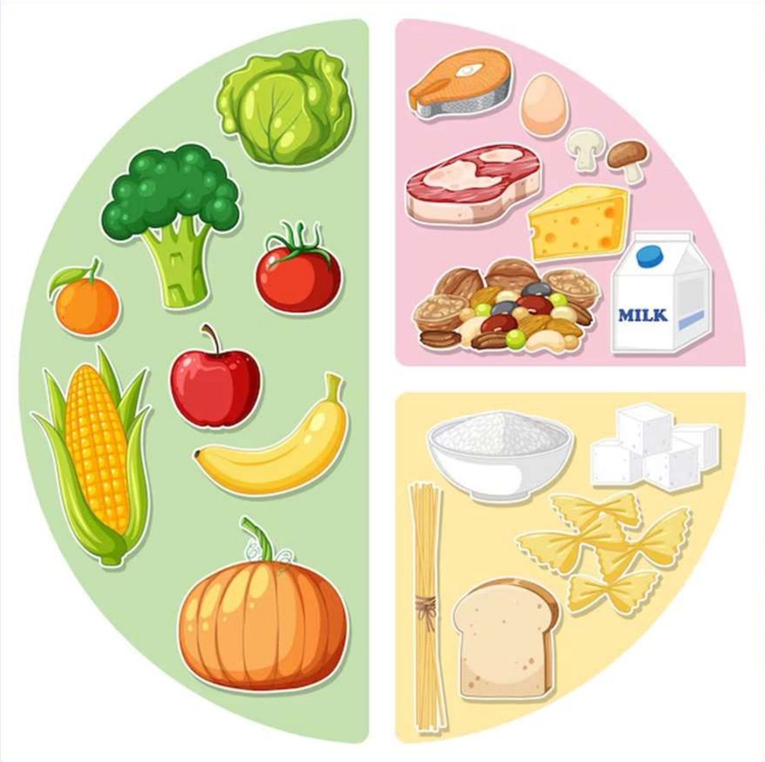
AHC Application 1 (Dalimunthe , 2021)

Goal:

- Make a system to sort foods by nutrition to help people manage their health better
- Help people plan healthier diets

How AHC was involved

- Calculate average nutrition value for different food groups
- Group the food based on similarity



Agglomerative Hierarchical Clustering (AHC)

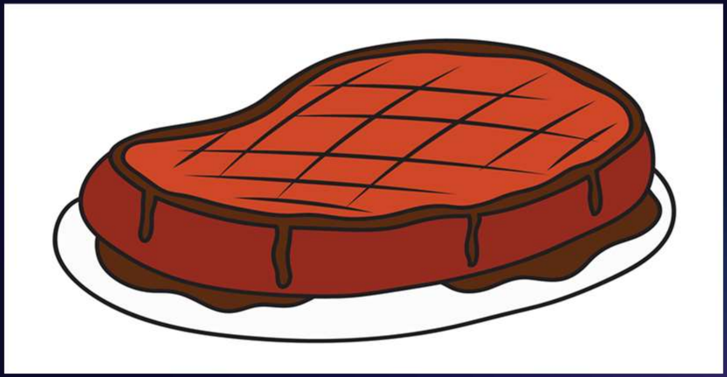
Application 2

Applying cluster analysis and preference mapping to assess consumer preference for different cooking temperatures of beef steaks

AHC Application 2 (Schmidt et al., 2010)

Goal:

- Understand consumer preference for steak cooked in different levels
- How these preference relate to each characteristics



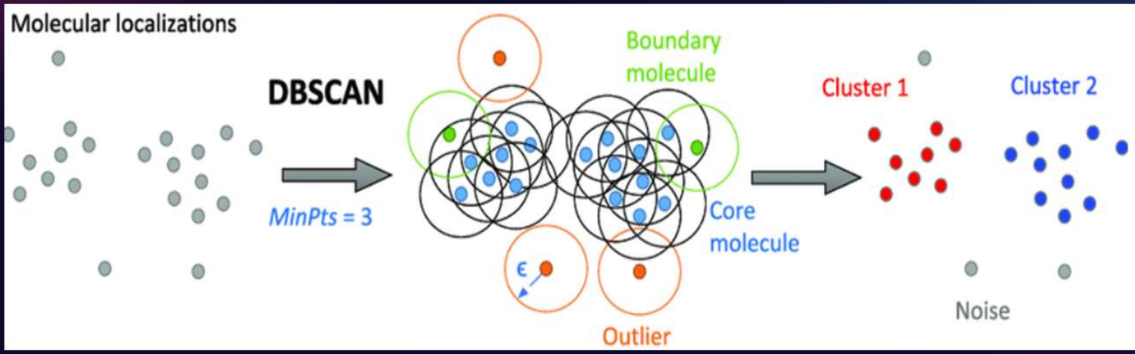
How AHC was involved

- Group consumer based on preference of steak
- Methods such as Ward’s method was used to measure the similarities between customer preferences

Algorithm 5: DBSCAN
(Density Based Clustering of Application with Noise)

DBSCAN

- Defines clusters as dense region and separates them by areas of low density
- Parameters:
  - Epsilon: radius of neighbourhood around the point
  - MinPts: Specifies the minimum number of points
- Important Steps:
  - Input Parameters
  - Identify Core Points
  - Cluster Expansion
  - Handle Border & Noise Points
  - Format Cluster
  - Output



DBSCAN

Application 1

Finding the Optimal Restaurant Location in Düsseldorf Germany

DBSCAN Application 1 (Xu, X., 2021)

Goal:

- To find the optimal location for opening a restaurant (area with high population density + minimal no. of nearby restaurants)

How DBSCAN was involved:

- Effective at identifying arbitrarily shaped clusters
- Produced 8 clusters & helped to identify outlier restaurants

DBSCAN

Application 2

Opening a new pizzeria in Turin using DBSCAN

DBSCAN Application 2 (Huseynov, A., 2021)

Goal:

- Determine the optimal location for new pizzeria
- Considering rental prices, space availability, low restaurant density, and high residential density

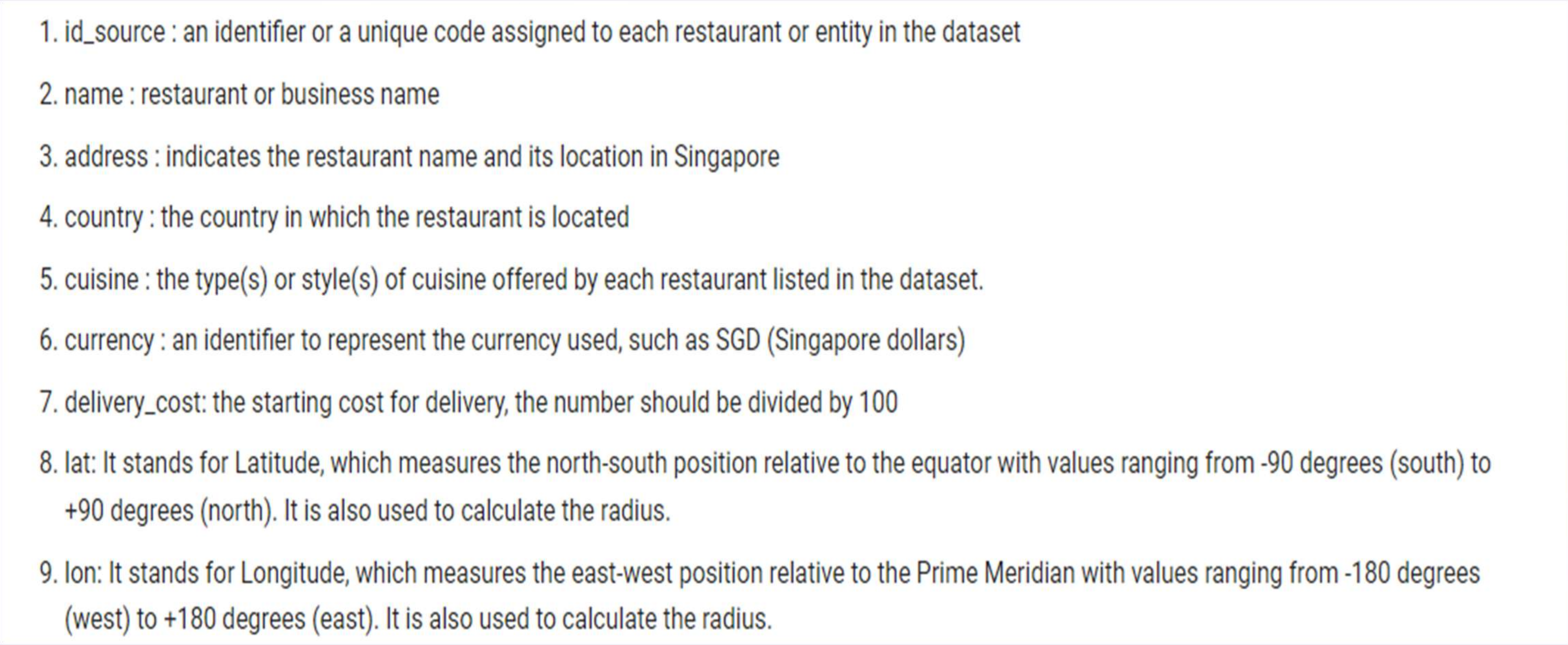
How DBSCAN was involved:

- Cluster zones based on proximity to nearby restaurants
- Ensure avoidance of duplication in the same category by clustering zones

3.0 EDA

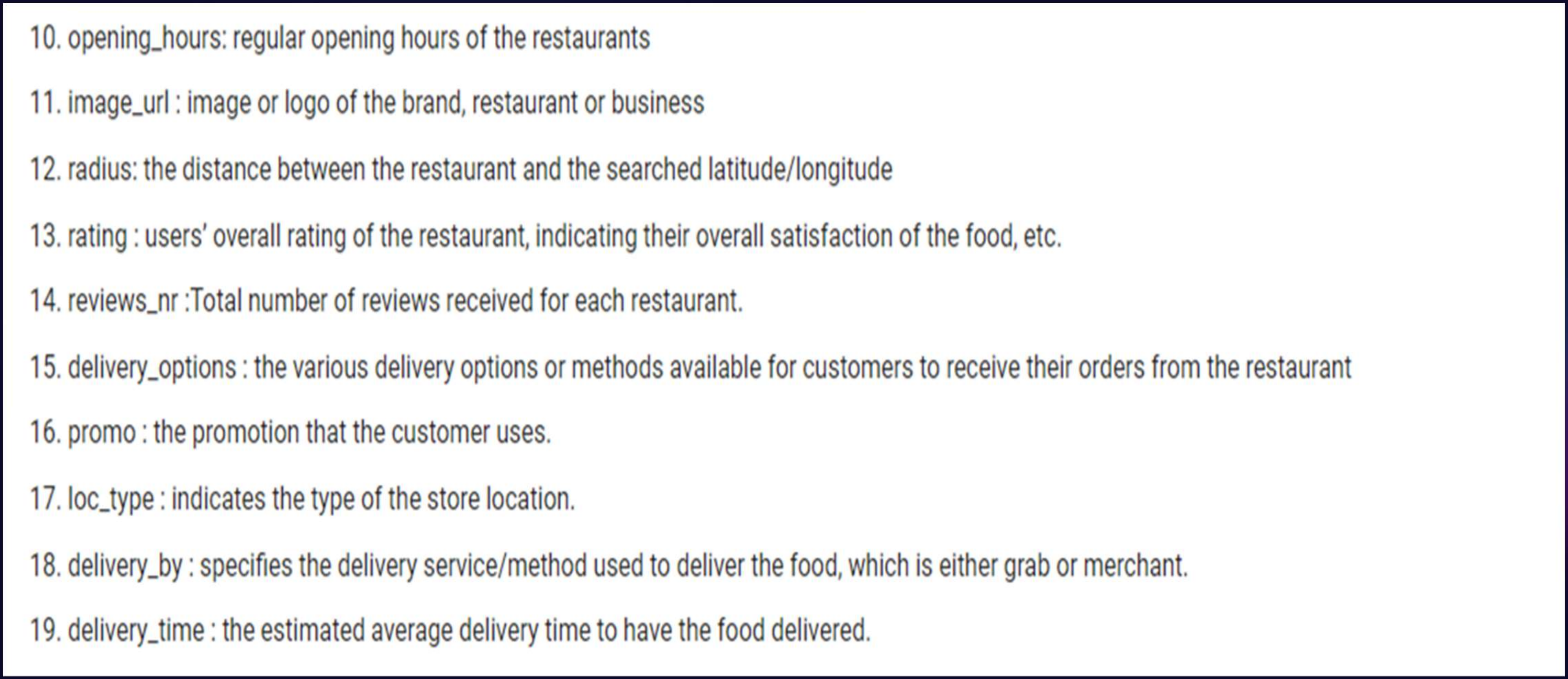
(See Code File)

Brief Overview of Dataset



1. id_source : an identifier or a unique code assigned to each restaurant or entity in the dataset
2. name : restaurant or business name
3. address : indicates the restaurant name and its location in Singapore
4. country : the country in which the restaurant is located
5. cuisine : the type(s) or style(s) of cuisine offered by each restaurant listed in the dataset.
6. currency : an identifier to represent the currency used, such as SGD (Singapore dollars)
7. delivery_cost: the starting cost for delivery, the number should be divided by 100
8. lat: It stands for Latitude, which measures the north-south position relative to the equator with values ranging from -90 degrees (south) to +90 degrees (north). It is also used to calculate the radius.
9. lon: It stands for Longitude, which measures the east-west position relative to the Prime Meridian with values ranging from -180 degrees (west) to +180 degrees (east). It is also used to calculate the radius.

Brief Overview of Dataset



10. opening_hours: regular opening hours of the restaurants

11. image_url : image or logo of the brand, restaurant or business

12. radius: the distance between the restaurant and the searched latitude/longitude

13. rating : users' overall rating of the restaurant, indicating their overall satisfaction of the food, etc.

14. reviews_nr :Total number of reviews received for each restaurant.

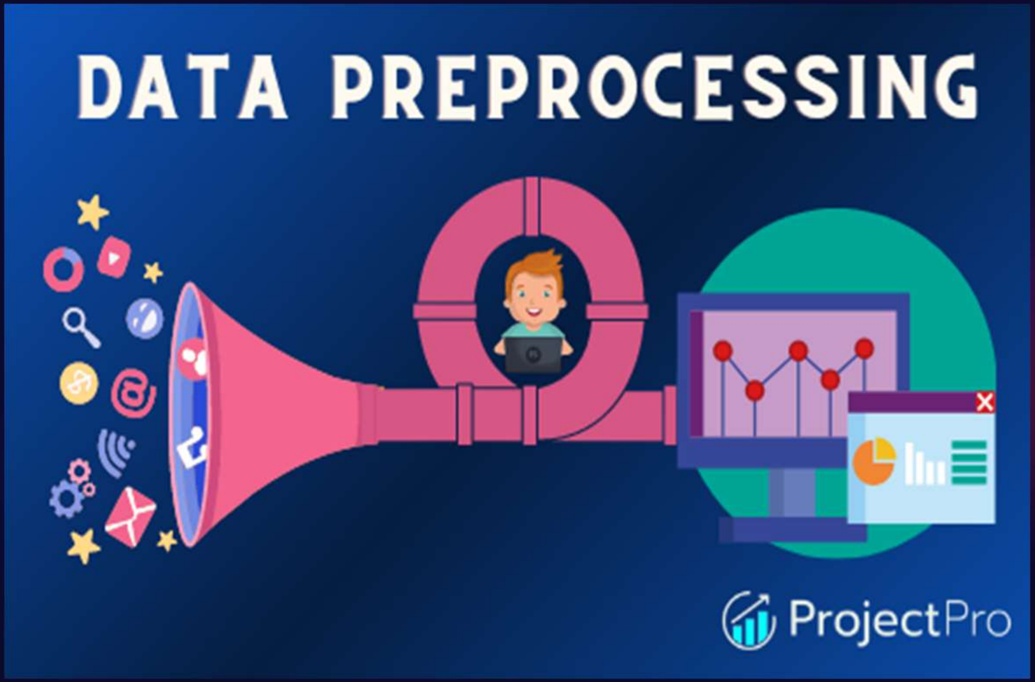
15. delivery_options : the various delivery options or methods available for customers to receive their orders from the restaurant

16. promo : the promotion that the customer uses.

17. loc_type : indicates the type of the store location.

18. delivery_by : specifies the delivery service/method used to deliver the food, which is either grab or merchant.

19. delivery_time : the estimated average delivery time to have the food delivered.



4.0 Data Preprocessing

(See Code File)

Data Preprocessing (UPDATED)

In summary, we did the following:

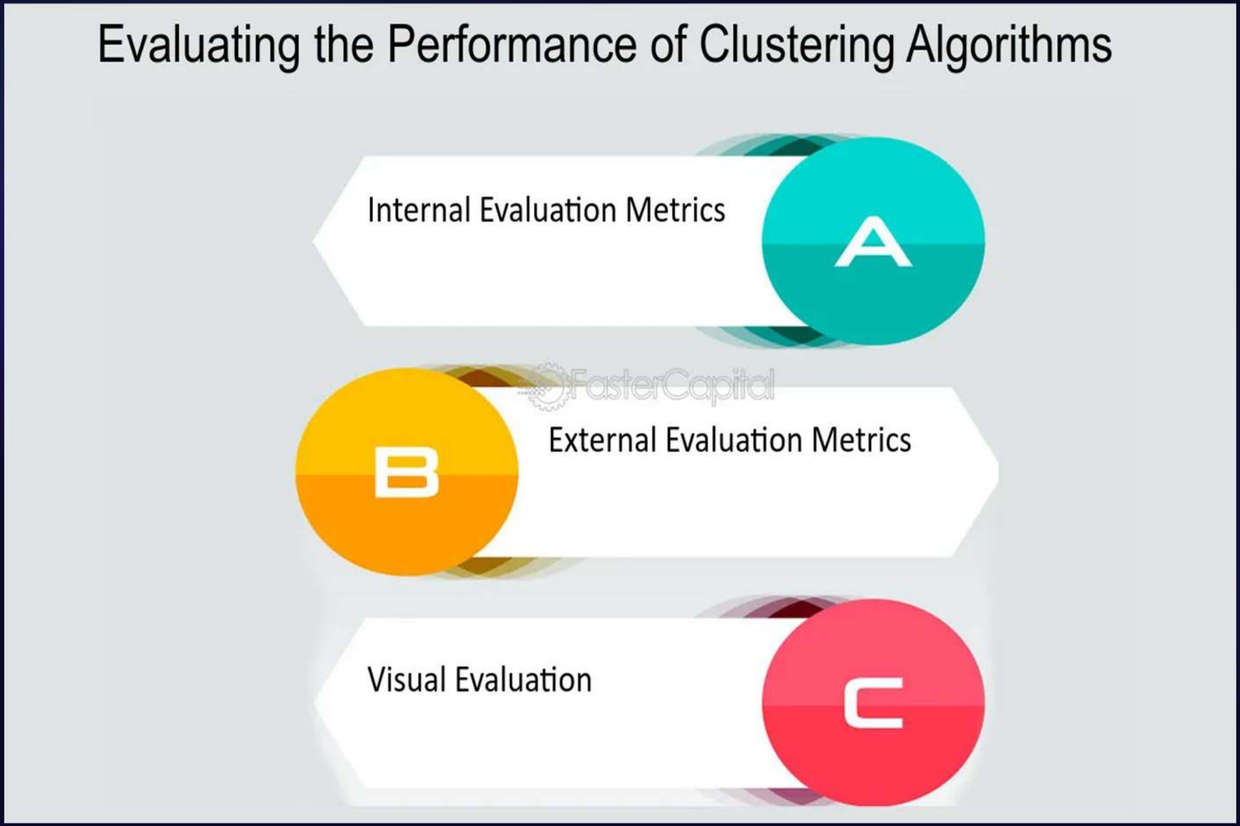
- Handled missing data
- Handled outliers
- Handled contaminated data
- Handled Inconsistent data
- Handled Invalid data
- Feature Engineering

We did not do the following because our dataset did not have those errors:

- Handling structural errors
- Handling duplicated data (our data has no duplicated records)

Assignment: Part 2

5.0 Performance Metrics Used



Evaluating the Performance of Clustering Algorithms

A Internal Evaluation Metrics

B External Evaluation Metrics

C Visual Evaluation

5.0 Performance Metrics Used

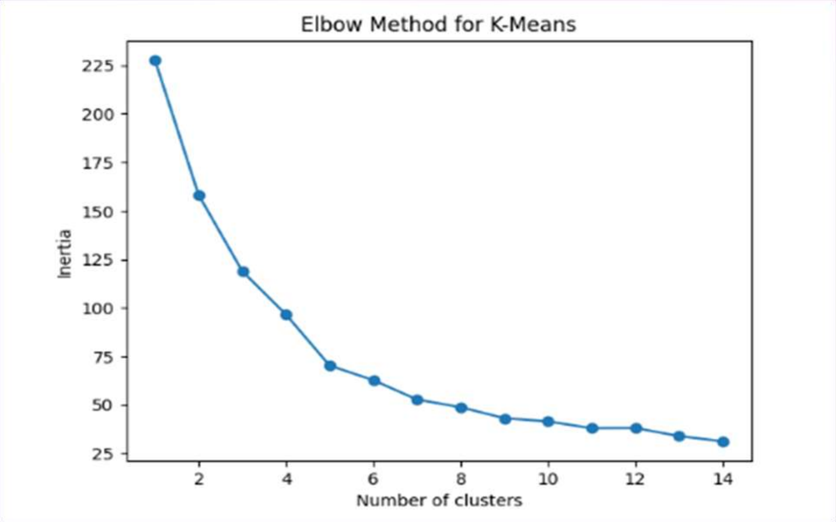
To Evaluate Models' Performance with Default or Fine-tuned Hyperparameter Values

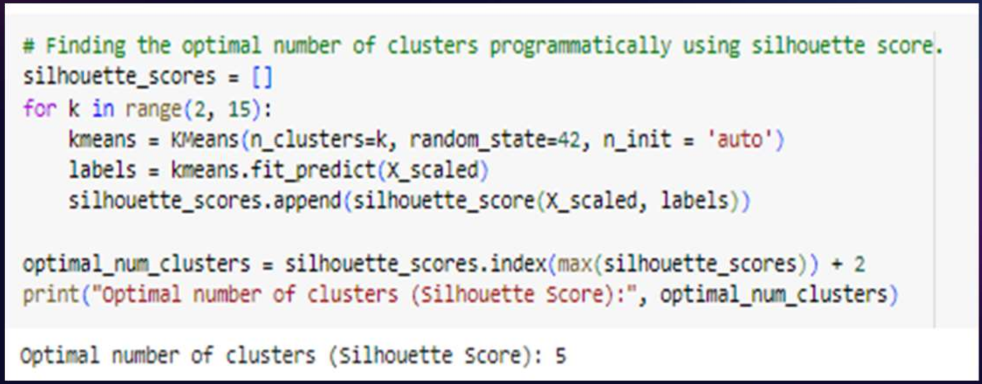
- **Silhouette Score**
  a. How similar an object is to its own cluster compared to other clusters
- **Davies-Bouldin Index**
  a. Average similarity between clusters.
- **Calinski-Harabasz Score**
  a. Based on the ratio of between-cluster dispersion and within-cluster dispersion
- **Dunn Index**
  a. Ratio between minimum inter-cluster distance & maximum intra-cluster distance
- **Hubert & Levin C index**
  a. How spread out clusters are compared to the total dispersion in a dataset

5.0 Performance Metrics Used (con't)

- **Determining Optimal K for K-Means using Elbow Method (Inertia) & Silhouette Score**
  a. Elbow Method (Inertia): Optimal K value is approximately 5, where the inertia begins decreasing linearly

  b. Silhouette Score Method: Optimal K value is 5.



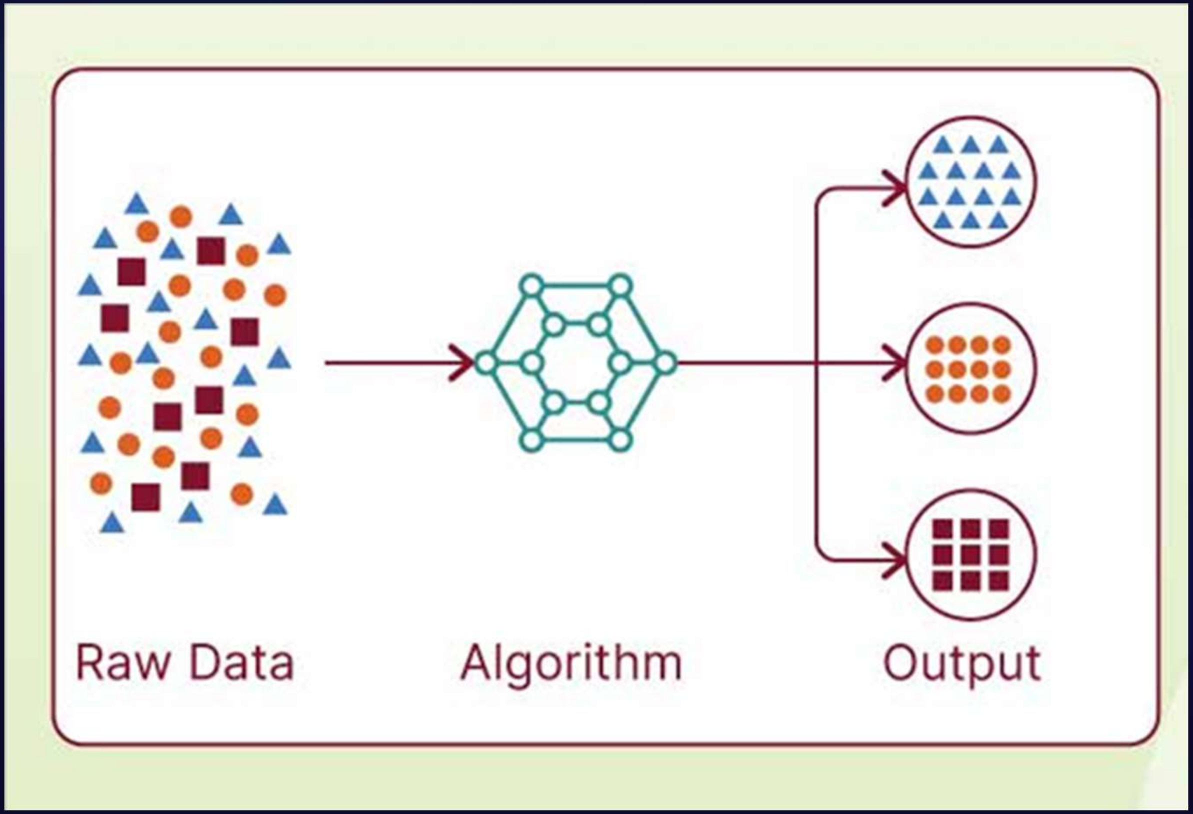


```
# Finding the optimal number of clusters programmatically using silhouette score.
silhouette_scores = []
for k in range(2, 15):
    kmeans = KMeans(n_clusters=k, random_state=42, n_init = 'auto')
    labels = kmeans.fit_predict(X_scaled)
    silhouette_scores.append(silhouette_score(X_scaled, labels))

optimal_num_clusters = silhouette_scores.index(max(silhouette_scores)) + 2
print("Optimal number of clusters (Silhouette Score):", optimal_num_clusters)
```

```
Optimal number of clusters (Silhouette Score): 5
```

6.0 Columns to be Clustered



6.0 Columns to be Clustered

- **3 Columns:**
  - Longitude
  - Latitude
  - Rating

- **Why?**
  - Aim: Find areas that have demand for Western food BUT the existing Western restaurants in those areas have Low Ratings.

- **Dimensionality reduction isn't needed:**
  - Clustering on just 3 columns (low-dimensional)
  - Already easy to analyse

7.0 Feature Scaling: MinMaxScaler

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

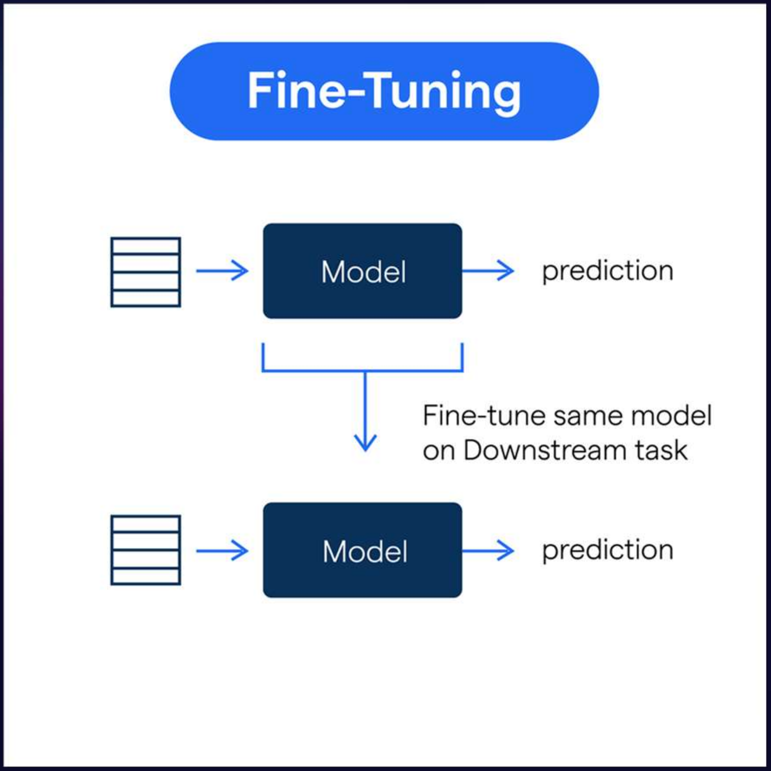
7.0 Feature Scaling: MinMaxScaler

Why?

- **Latitude** and **longitude** fall within **specific ranges,** making MinMaxScaler ideal for preserving their spatial relationships.

- **Rating** values **(specific range: 1 to 5)** will be normalised to maintain their relative differences across the dataset.

8.0 Default & Fine Tuned Models



8.1 Default Models

- **K-Means**
  - Initialized with default hyperparameters (n_clusters = 5).
- **BIRCH**
  - Initialized with default hyperparameters (threshold=0.1).
- **Affinity Propagation**
  - Initialized with default hyperparameters (damping=0.5, preference=None).
- **Agglomerative Hierarchical Clustering**
  - Initialized with default hyperparameters.
- **DBSCAN Clustering**
  - Initialized with default hyperparameters (eps=0.1, min_samples=7).

* Evaluated their performance using the 5 Performance Metrics on Slide 48

8.2 Fine Tuned Models

- **K-Means Clustering**

- Utilized GridSearchCV to find optimal hyperparameters (init, max_iter, tol).

- **BIRCH Clustering**

- Employed GridSearchCV to find the optimal hyperparameters (threshold, branching_factor, n_clusters).

- **Affinity Propagation Clustering**

- Used RandomizedSearchCV to search for optimal hyperparameters (damping, preference).
- Adjusted the preference parameter iteratively to achieve a reasonable number of clusters.

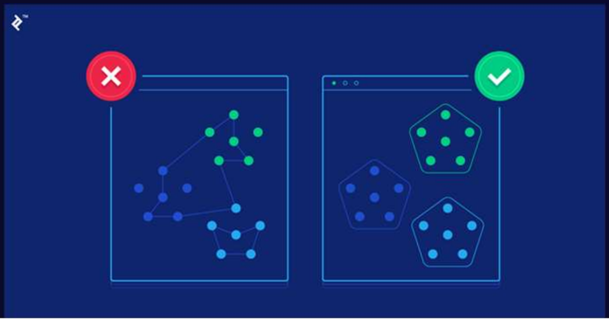
8.2 Fine Tuned Models

- **Agglomerative Hierarchical Clustering**

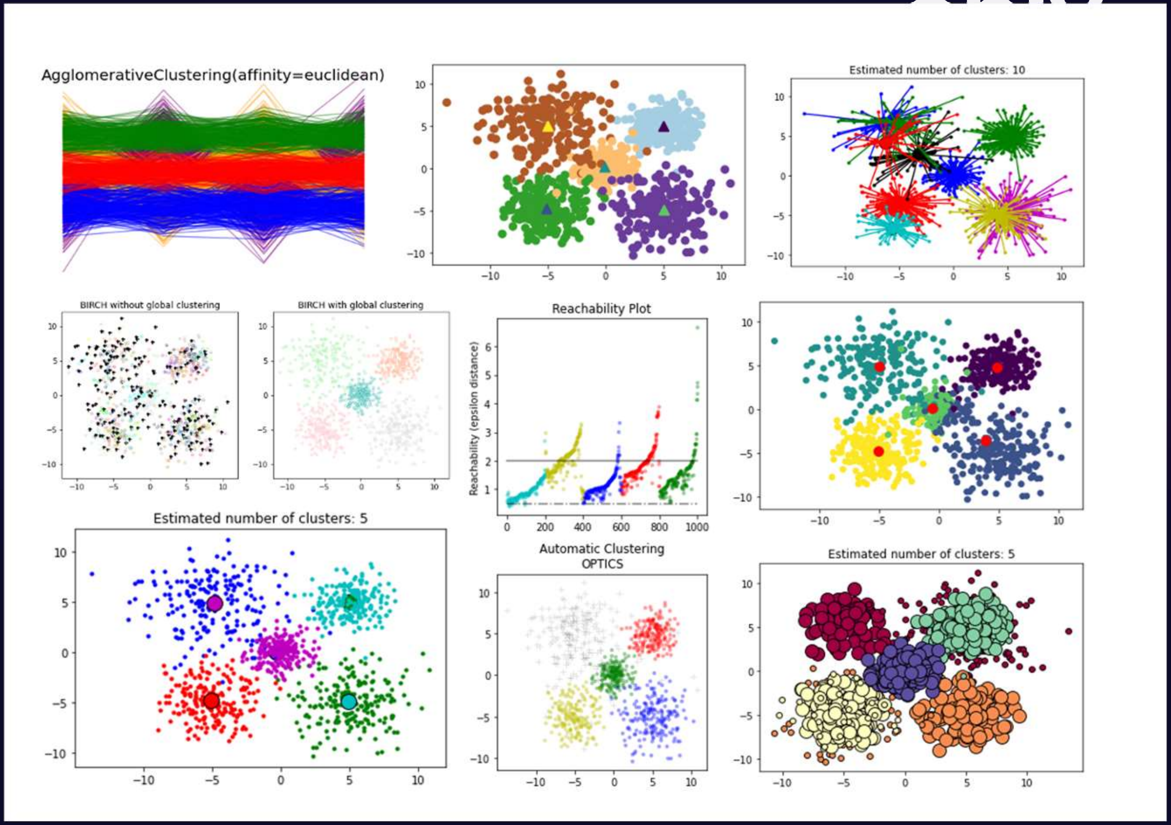
- Used a nested for loop to search for optimal hyperparameters (n_clusters, affinity, linkage).
- Cannot use GridSearchCV and RandomizedSearchCV directly to fine-tune it

- **DBSCAN Clustering**

- Utilized GridSearchCV to find optimal hyperparameters (eps, min_samples).
- Defined a parameter grid and performed a systematic search.
- Selected the best hyperparameters and evaluated the model's performance.



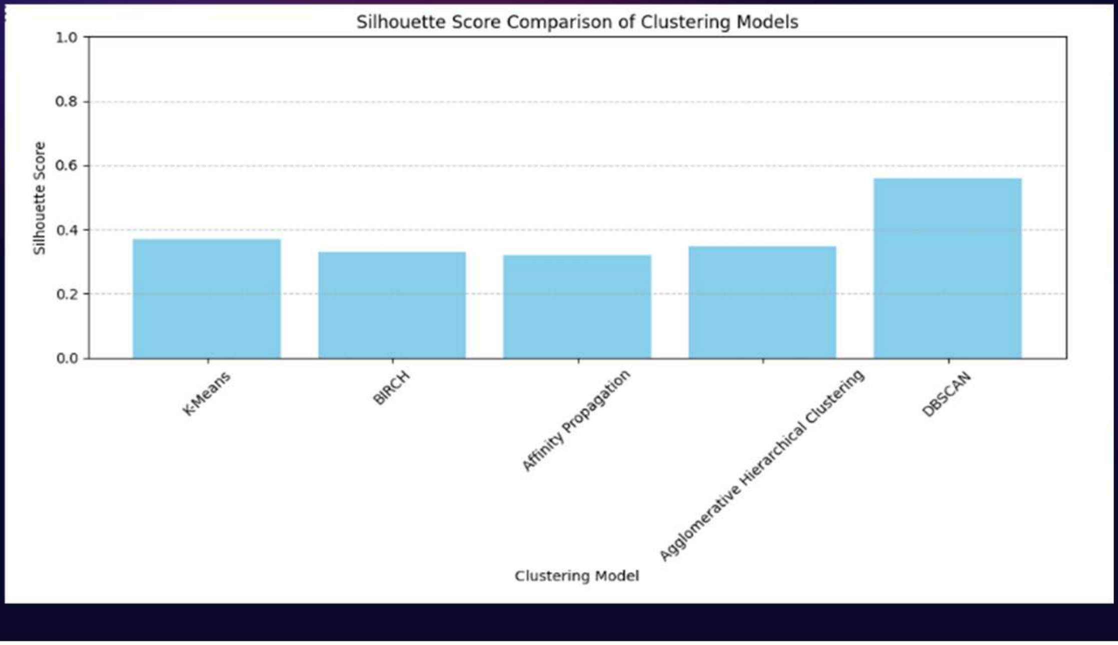
9.0 Performance of Fine-Tuned Models only



Note:
Comparisons of Performance between Default & Fine-tuned Models can be found in the coding file

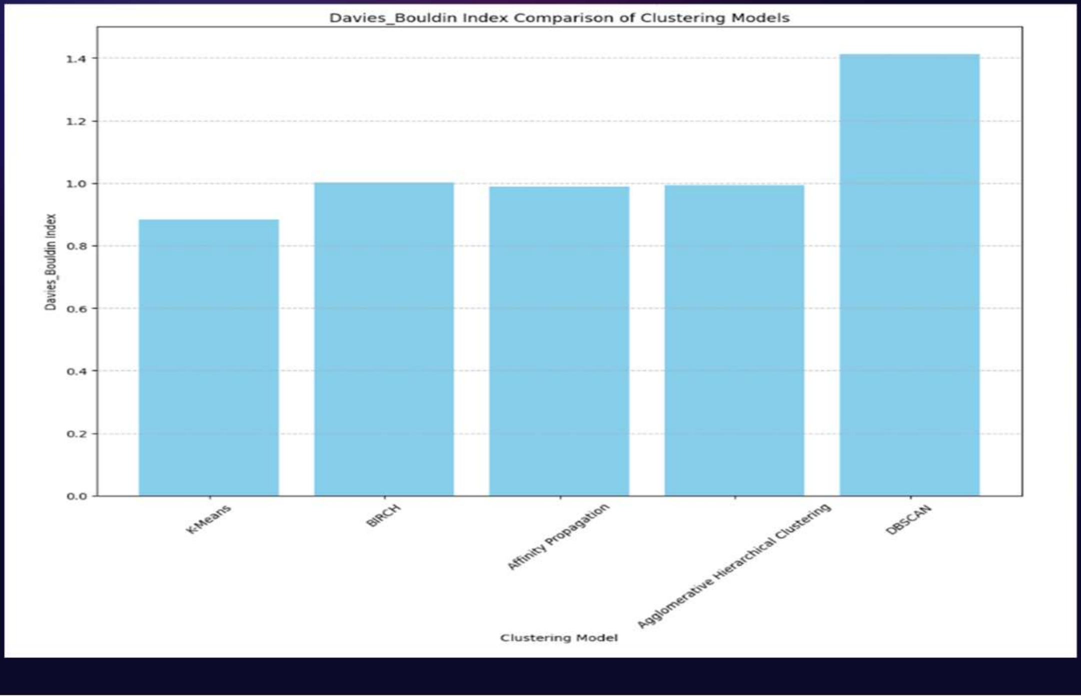
Silhouette Score (Fine-tuned Models only)

- DBSCAN has the highest silhouette score (points within clusters are densely packed & well-separated from points in other clusters).
- The other 4 models have roughly the same silhouette score
- Affinity Propagation has the lowest silhouette score.



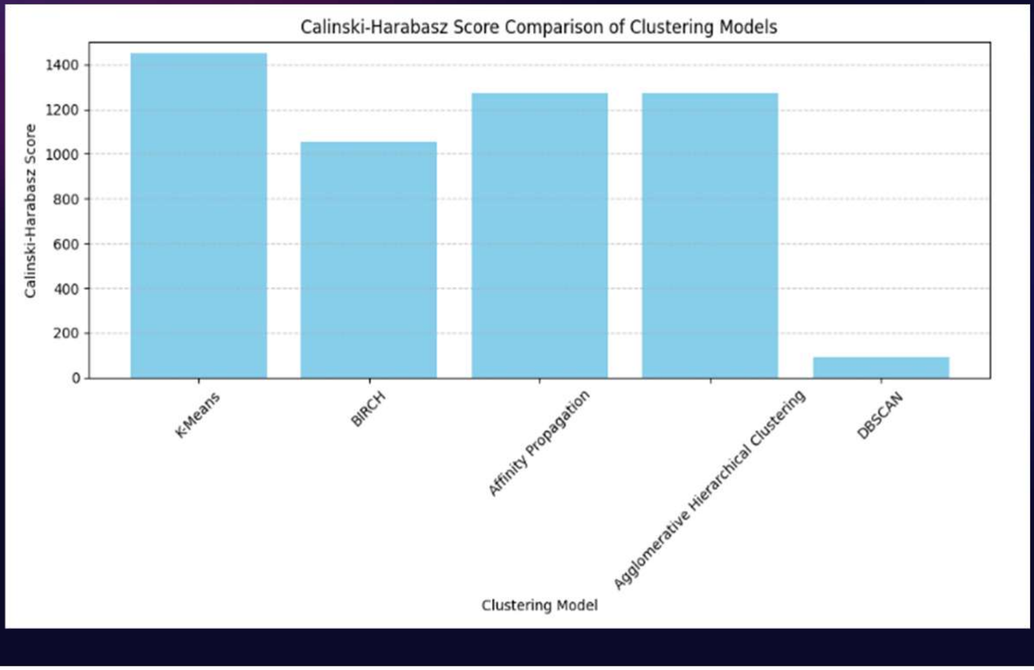
Davies-Bouldin Index (Fine-tuned Models only)

- DBSCAN has the highest Davies-Bouldin index (clusters are less well-separated or more dispersed).
- BIRCH, Affinity Propagation and AHC have approximately the same score
- K Means has the lowest Davies-Bouldin index (better defined clusters).



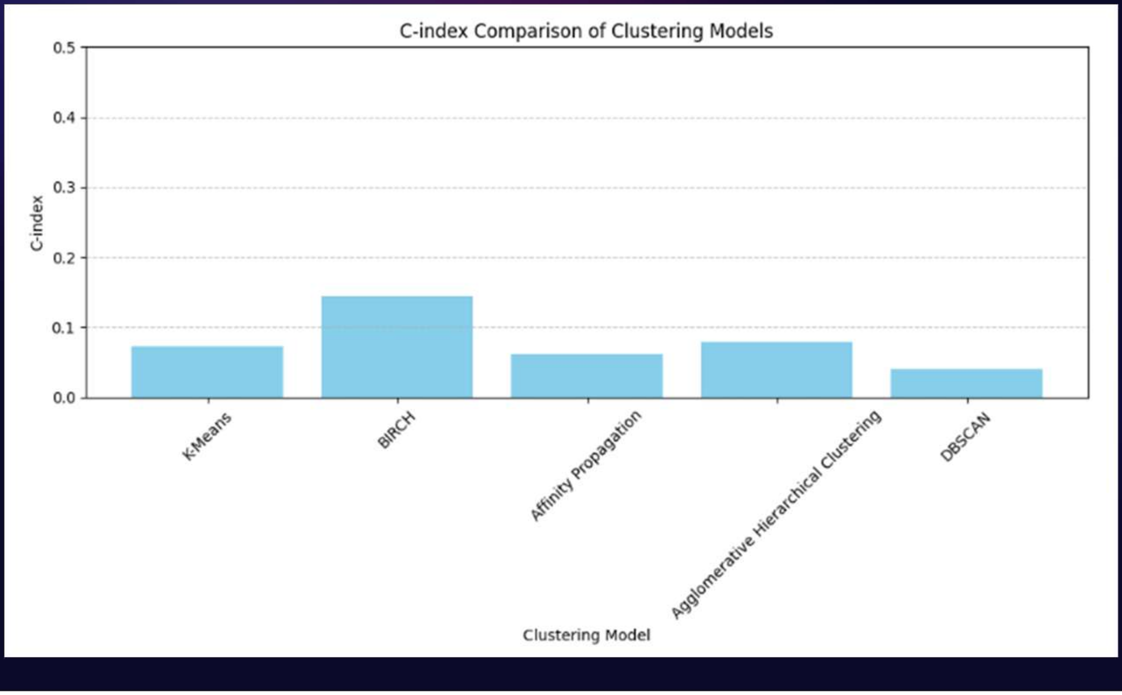
Calinski-Harabasz (Fine-tuned Models only)

- K Means achieves the highest Calinski-Harabasz score (well-defined clusters with instances closely grouped within each and distant from instances in other clusters).
- Affinity Propagation and AHC score similarly.
- Birch outperforms DBSCAN, which fails to effectively separate clusters, leading to significant overlap.



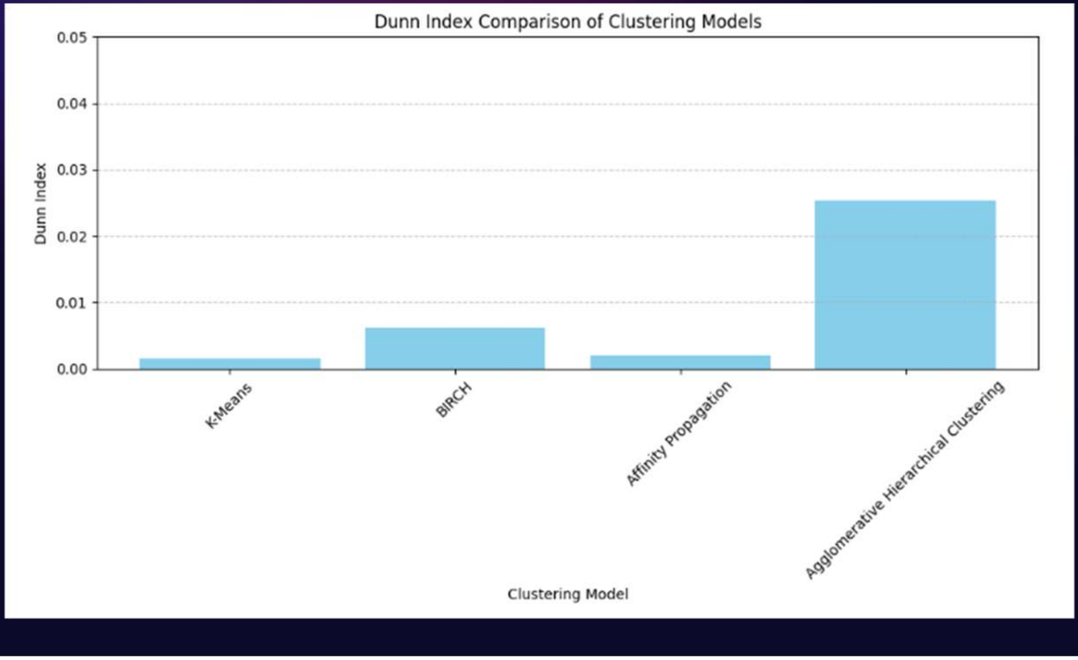
Hubert & Levin C-index (Fine-tuned Models only)

- DBSCAN has the lowest Hubert & Levin C-index (produced the most optimal number of clusters, 3), followed by Affinity Propagation, K Means and AHC.
- Birch has the highest index, indicating that its internal quality index is the worst.



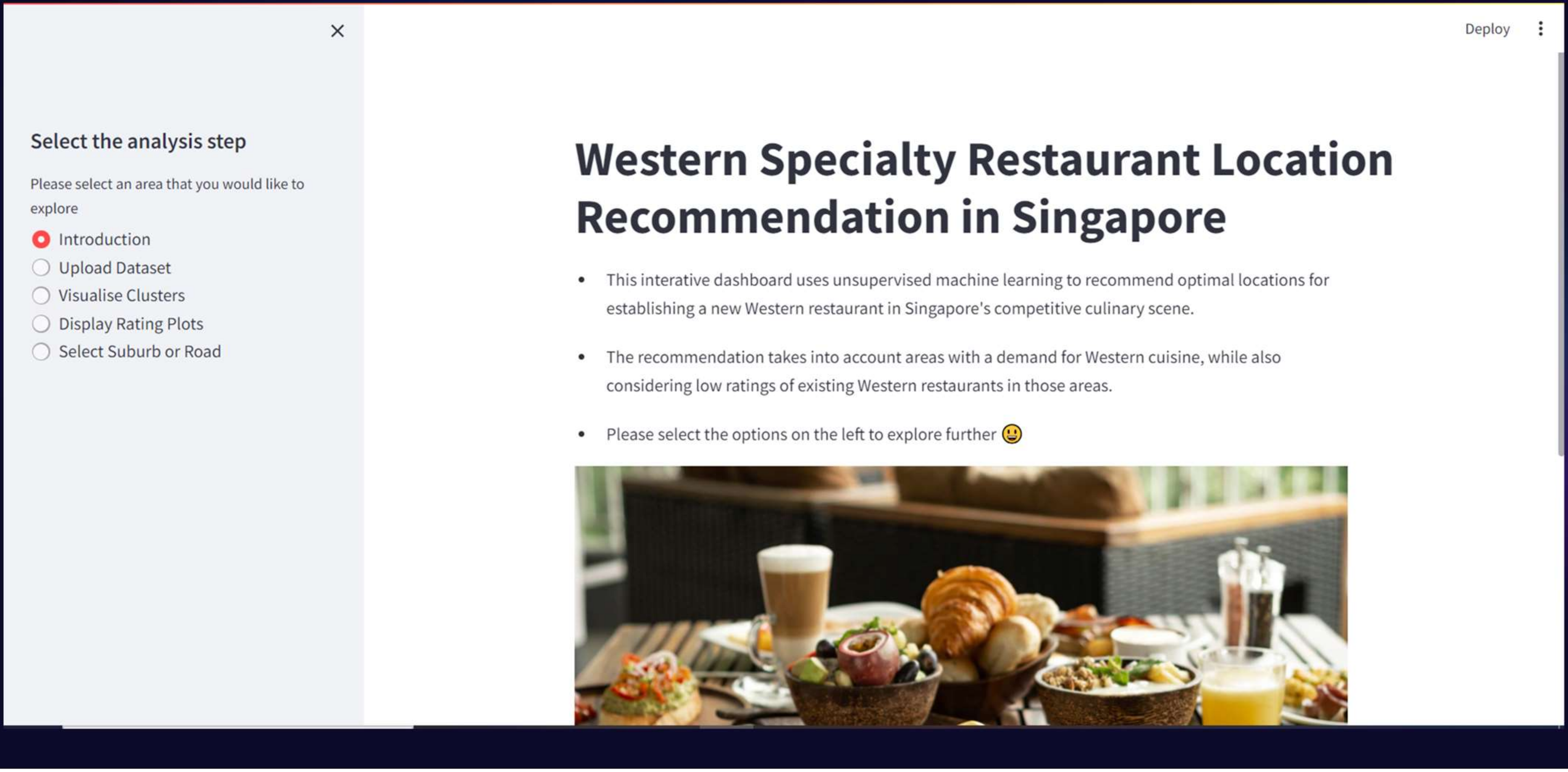
Dunn Index (Fine-tuned Models only)

- AHC has the highest Dunn Index (its clusters are the most compact & well-separated), followed by Birch and Affinity Propagation.
- K Means, on the other hand, has the lowest Dunn Index, suggesting poorer clustering quality.
- The Dunn Index isn't applicable to DBSCAN due to its assumption of clear cluster boundaries (doesn't align with the output of density-based clustering).



10.0 Location Recommendation using Streamlit

Our Dashboard



Select the analysis step

Please select an area that you would like to explore

● Introduction
○ Upload Dataset
○ Visualise Clusters
○ Display Rating Plots
○ Select Suburb or Road

Western Specialty Restaurant Location Recommendation in Singapore

- This interative dashboard uses unsupervised machine learning to recommend optimal locations for establishing a new Western restaurant in Singapore's competitive culinary scene.
- The recommendation takes into account areas with a demand for Western cuisine, while also considering low ratings of existing Western restaurants in those areas.
- Please select the options on the left to explore further 😃

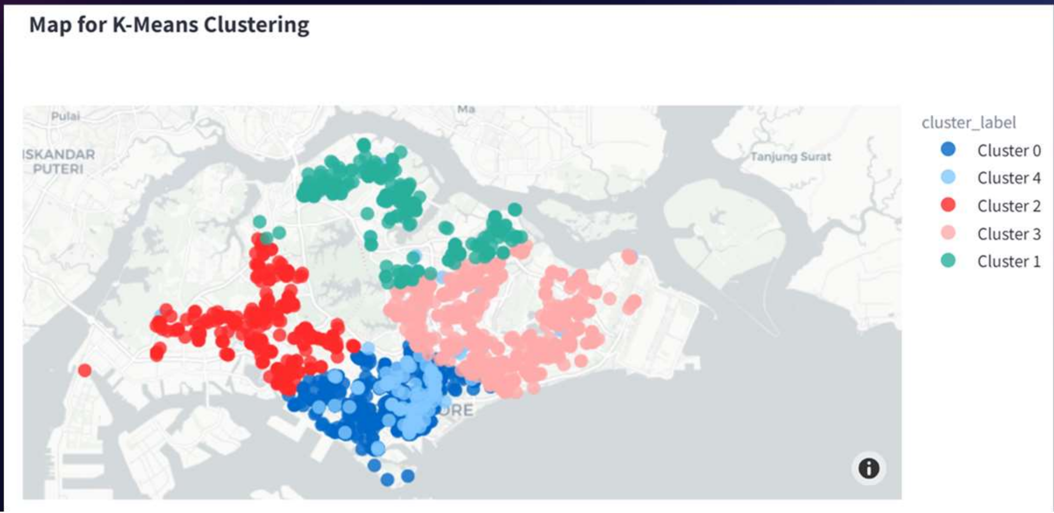
K-means clustering map

Observation

- Dark blue cluster has the highest avg rating (south side of the map).
- Light blue cluster has the lowest avg rating (same location with dark blue cluster).
- Green cluster has the second lowest avg rating (top north side of the map)

Recommendation

- The ideal location is at the north side of Singapore
- Located at the green cluster side


Map for K-Means Clustering

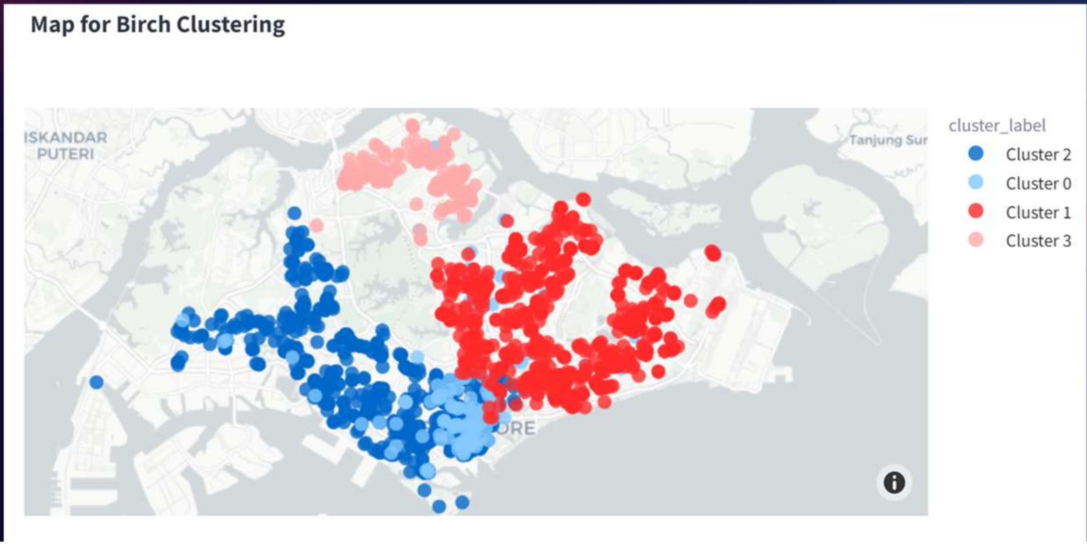
Birch clustering map

Observation

- Dark blue cluster has the highest avg rating (west side of the map).
- Light blue cluster has the lowest avg rating (south side).
- Beige cluster has the second lowest avg rating (top north side of the map)

Recommendation

- The ideal location is at the north side of Singapore
- Located at the beige cluster side



Map for Birch Clustering

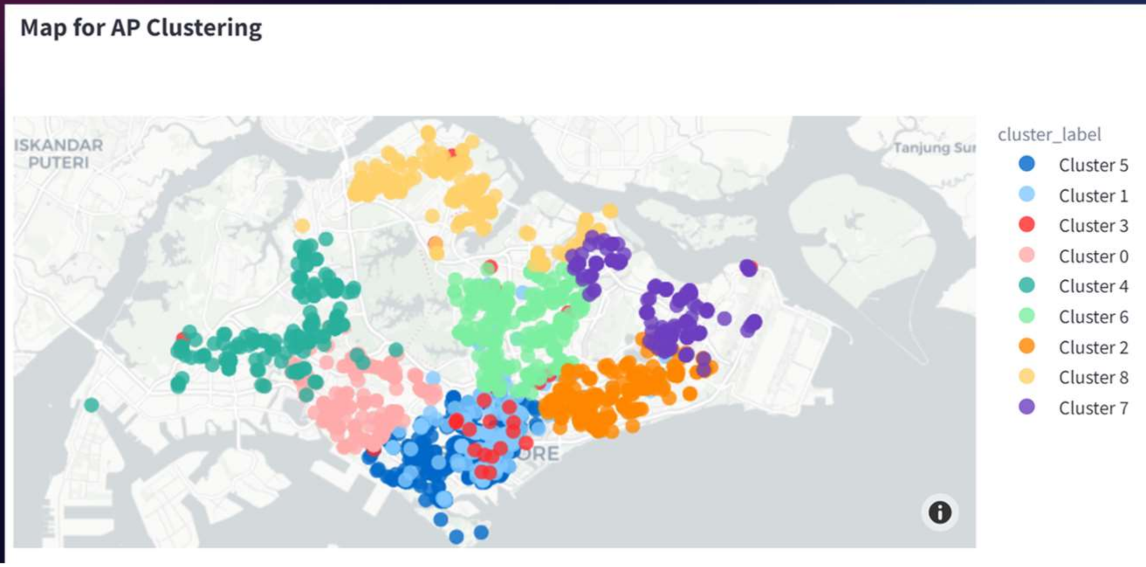
AP clustering map

Observation

- Dark blue cluster has the highest avg rating (south side of the map).
- Yellow cluster has the 4th lowest avg rating (north side).
- Light blue and red clusters have very low ratings (south side of the map)

Recommendation

- Ideal location is at the north side of Singapore
- Located at the yellow cluster side



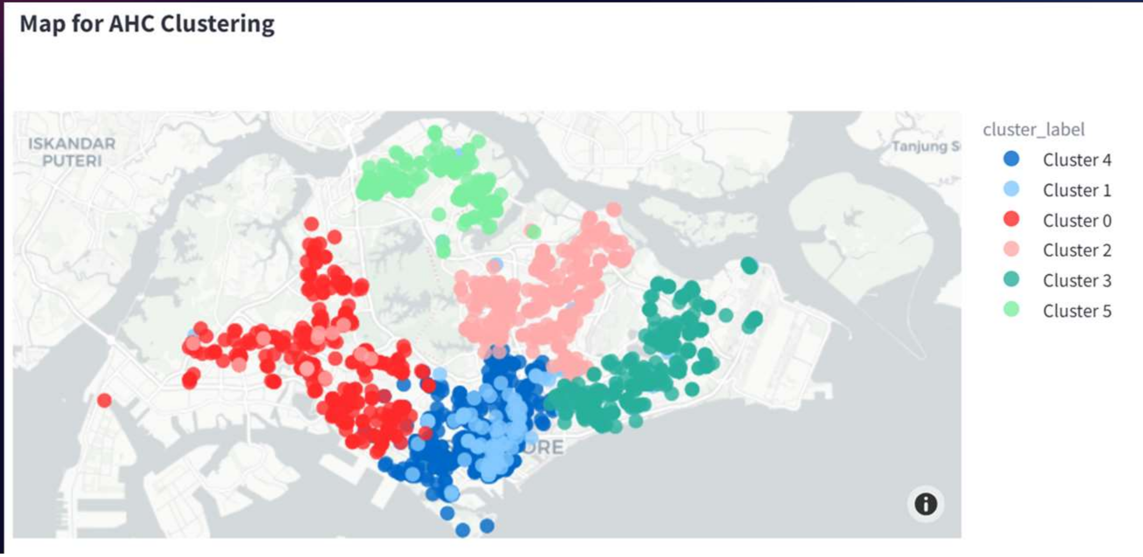
AHC clustering map

Observation

- Red cluster has the highest avg rating (south side of the map).
- Light green cluster has the 3rd lowest avg rating (north side).
- Light blue cluster has the lowest avg rating (south side of the map)

Recommendation

- Ideal location is at the north side of Singapore
- Located at the light green cluster side



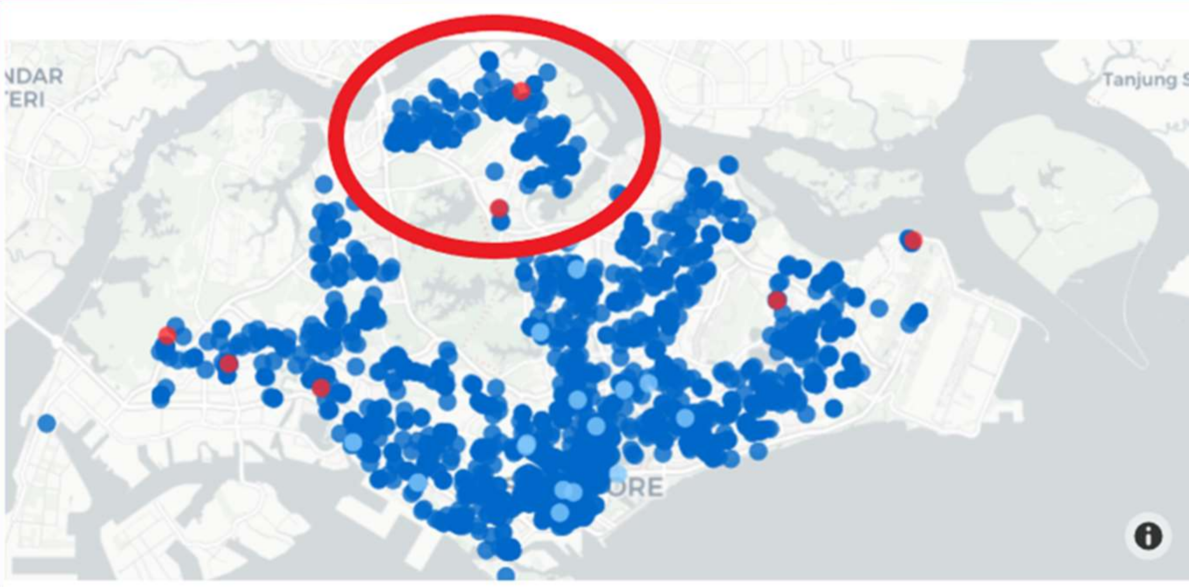
DBSCAN clustering map

Observation

- Majority of the points belong to dark blue cluster (hard to recommend)
- We exclude the DBSCAN cluster map from location recommendation

Recommendation

- Based on the other 4 clustering maps, opening the new restaurant at the top northern side is the most ideal place.



Thank
You