

International Journal Research Publication Analysis

Page: 01-15

STORE LOCATION OPTIMIZATION USING DATA ANALYTICS

***Kokne Sushmitha, Dr. Ramchandrar (Guide)**

Department of MCA (Master of Computer Applications) Chaitanya Bharathi Institute of Technology (CBIT), Gandipet, Hyderabad, Telangana, India.

Article Received: 05 September 2026

Article Revised: 25 September 2026

Published on: 15 October 2026

***Corresponding Author: Kokne Sushmitha**

Department of MCA (Master of Computer Applications) Chaitanya Bharathi Institute of Technology (CBIT), Gandipet, Hyderabad, Telangana, India.

DOI: <https://doi-doi.org/101555/ijrpa.6009>

ABSTRACT

Choosing where to open new stores is often settled with spreadsheets, rules of thumb or proprietary models whose data preparation and evaluation are rarely documented. This paper documents a small, reproducible analytics workflow for store location optimization and applies it to a synthetic city of 400 one-square-kilometre zones, 40 candidate sites and 12 competitor stores. Python and pandas derive demand and rent indicators, the tables are stored in SQLite and queried with SQL, k-means segments the zones, and six selection methods (population rank, weighted score, k-means snapping, maximal covering by mixed-integer programming, greedy-plus-swap search on a Huff-model profit, and exhaustive search) each choose 5 sites. All six are scored under one Huff-model profit function and then re-scored under five distance-decay values and 500 Monte Carlo draws. In the demonstration data the Huff-based search reaches a net annual profit of INR 2.67 crore, against 1.83–2.31 crore for the four baselines and 1.18 crore for a random five-site choice. Greedy-plus-swap matches the exhaustive optimum among all 658,008 five-site combinations, the covering model attains the highest demand coverage (52.2%) without the highest profit, and baseline regret grows with distance decay. Because the data are synthetic and the Huff-based methods are judged under their own objective, the results show how the workflow behaves, not how a real retail market behaves.

INDEX TERMS: Store Location, Site Selection, Retail Analytics, Huff Model, Maximal Covering, K-Means, Sqlite, Python.

I. INTRODUCTION

Where a store stands shapes how many people can reach it, how many rivals share its customers and how much rent it must earn back. The decision is costly to reverse, and the literature on retail location has long treated it as a modelling problem in its own right [1]. Firms weighing a market entry may face hundreds of dispersed candidate locations and little store-level information, which makes the first screening step hard to do well by hand [2].

Several practical questions follow. Where is demand concentrated? How strongly do existing competitors pull customers away from a given site? Which set of sites, taken together, earns the most, given that two stores of the same chain in one neighbourhood split the same customers? Ranking candidates one by one answers the first two questions but not the third, because the value of a site depends on which other sites are chosen.

Python suits the preparation and analysis stages. The pandas library provides labelled tables for cleaning and aggregation [3], NumPy supplies array routines [4], scikitlearn offers clustering [5], SciPy includes a mixed-integer solver [6], and Matplotlib draws the figures [7]. SQLite keeps a whole database in one file, which is enough for a project that still benefits from structured queries.

The research problem has two parts. First, how can a zonelevel demand table and a list of candidate and competitor sites be cleaned, stored, queried and turned into a set of recommended store locations in a way that others can reproduce? Second, when several common selection methods are compared on the same footing, how much does the choice of method matter for profit, demand coverage, cannibalisation and robustness to uncertain parameters?

The contributions are modest but concrete. (i) A documented pipeline turns a raw CSV of zones, candidates and competitors into a SQLite database and a ranked set of sites. (ii) Six selection methods, from a population ranking to an exhaustive search, are implemented and scored under one Huff-model profit function. (iii) The greedy-plus-swap heuristic is checked against the exhaustive optimum over all five-site combinations. (iv) The methods are re-scored under five values of the distance-decay exponent and 500 Monte Carlo draws. (v) A worked analysis of a synthetic city is reported together with the limits of what such data can support.

The rest of the paper is organised as follows. Section II reviews related work, Section III states the research gap and problem, and Section IV lists the objectives. Sections V to VIII describe the dataset, methodology, architecture and implementation. Results are given in Section IX, followed by discussion, limitations, future work and conclusions.

II. RELATED WORK

Classical location theory supplies the vocabulary. Hotelling analysed two sellers choosing positions on a line and showed how competition pulls them together [8]. Huff modelled the probability that a consumer patronises a store as increasing in its attractiveness and decreasing in distance, relative to all alternatives [9]. On the optimisation side, Hakimi formulated the median problem on a network [10], and Church and ReVelle introduced the maximal covering location problem, which selects a fixed number of facilities to cover as much demand as possible [11]. ReVelle and Eiselt survey the field [12], and Craig *et al.* review the models used specifically for the retail location process [1].

Applied studies combine such ideas with spatial data. Roig-Tierno *et al.* combined geographic information systems with the analytical hierarchy process to structure the site decision for a supermarket in a Spanish city [13]. Karamshuk *et al.* mined data from online location-based services to study optimal retail store placement [14]. More recently, Lu *et al.* built a screening model for a jewellery chain from public city-, competitor- and point-of-interest data, using 743 existing stores, and found that a sequential ensemble model performed best [2]. Gao *et al.* applied gradient-boosted trees with interpretation methods to the locations of two coffee chains in China and reported that traditional and new retail respond differently to population density, rent and competition [15].

Taken together, the optimisation papers concentrate on formulations, the machine-learning papers on predicting store performance from data that is often proprietary, and the GIS studies on decision structure for a single case. Table I summarises the comparison.

Table I: Comparison With Related Approaches.

Work	Data and method	Main output
[1]	Review of retail location process models	Survey
[8]	Two sellers on a line	Competitive equilibrium
[9]	Probabilistic trade-area model	Patronage probabilities
[10]	Median problem on a network	Sites minimising distance
[11]	Maximal covering, integer programming	Demand covered by p sites
[13]	GIS with AHP, supermarket case	Site decision process
[14]	Location-based service data mining	Store placement study
[2]	743 jewellery stores; public city, competitor and POI data; ensemble ML	Site screening by predicted performance
[15]	Two coffee chains in China; XGBoost Location determinants with interpretation	
This work	Synthetic city; pandas, SQLite, six evaluation and method comparison	Documented workflow selection methods, Huff

III. RESEARCH GAP AND PROBLEM STATEMENT

Among the sources examined for this review, none described a compact, rerunnable workflow that starts from a raw zone-level table, passes it through documented cleaning and a SQL database, applies several selection methods of increasing sophistication, and scores all of them under one profit model with an exact benchmark and an uncertainty check. The observation is limited to the literature located here, which was not a systematic search, so other work may exist. It is also a statement about scope and not about quality: the studies above answer different questions.

Problem statement. Given a table of city zones with population, income, footfall and rent, a list of candidate sites and a list of competitor stores, design and document a reproducible workflow that selects a fixed number of new store sites, compare several selection methods under a common profit model, and state plainly what the data used can and cannot show.

IV. OBJECTIVES

The work pursued six objectives. (i) Clean and validate the zone, candidate and competitor tables and store them in SQLite. (ii) Derive annual demand and annual rent indicators. (iii) Write SQL queries for city totals, top zones and candidate catchments. (iv) Segment the zones by k-means. (v) Implement six site-selection methods and compare them on profit, coverage and cannibalisation. (vi) Test the methods against distance-decay values and Monte Carlo parameter draws.

V. DATASET AND DATA PREPROCESSING

A. Dataset Description

The project generates its own data. A script with a fixed random seed (2026) creates a hypothetical city of 20×20 zones of one square kilometre each, with a central business district and two sub-centres. Each zone has a population, an income index, a footfall index and a monthly rent per square foot (Tables II and III). Twelve competitor stores and forty candidate sites are placed in zones with probability increasing in footfall. The data are synthetic and describe no real city, so they are a test bed for the workflow and not evidence about any market. The tables hold aggregate counts only and contain no personal information.

Table II: Dataset Characteristics.

Property	Value
Zones	400 (20 × 20 grid, 1 km ² each)
Population	1,499,993 (per zone 306–12,615)
Income index	mean 1.09 (0.50–1.81)
Footfall index	20.0–100.0
Rent (INR/sq ft/month)	23.9–151.5, mean 61.7
Annual demand	INR 143.0 crore in total
Candidates / competitors	40 / 12
Missing / duplicate rows	0 / 0
Source	Synthetic, seed 2026

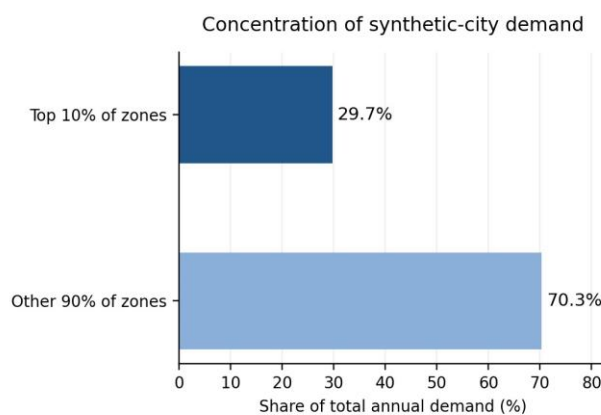


Fig. 2. Demand concentration across the 400 synthetic zones.

B. Data Quality

Checks on the raw table found no missing values, no duplicate rows and no value that failed numeric conversion.

Because the data were generated, these checks say more about the pipeline than about data quality in the field. Demand is concentrated: the 40 zones with the highest demand (10% of zones) hold 29.7% of the total. Footfall, rent and population are strongly related (Pearson $r = 0.97$ for footfall and rent, 0.95 for population and demand), but these relationships were built into the generator and should not be read as findings.

C. Data Preprocessing

Preprocessing is a function in the pipeline script. It coerces the numeric columns, drops duplicate and incomplete rows (none on this dataset) and derives two quantities. Annual demand of zone i in INR crore is

$$D_i = P_i \cdot s \cdot f_i \times 10^{-7} \quad (1)$$

where P_i is the population, I_i the income index and $s = \text{INR } 800$ the annual category spend per person at income index 1. Annual rent of a store at site j in INR crore is

$$K_j = r_j \cdot a \cdot 12 \times 10^{-7} \quad (2)$$

with r_j the rent per square foot per month and $a = 2,000$ sq ft the store area. One crore is 10^7 INR. Both quantities are model outputs, not observed figures. The parameters in Table III were chosen so that store revenue and profit fall in a plausible range; they were not fitted to data.

Table III: Model Parameters and Assumptions.

Parameter	Value
Stores to open, p	5
Distance decay, λ	2.0 (1.0–3.0 in sensitivity tests)
Attractiveness: new / competitor	1.2 / 1.0–1.6
Outside-option utility, u_0	0.60
Category spend, s	INR 800 per person per year
Contribution margin, m	0.18
Fixed cost per store, F	INR 0.45 crore per year
Store area, a	2,000 sq ft
Coverage radius, R_c	2.5 km
Score weights (demand, footfall, competition, rent)	0.40, 0.25, 0.20, 0.15

VI. PROPOSED METHODOLOGY

The workflow has seven stages: data generation or collection, cleaning, transformation, database storage, SQL analysis, segmentation and site selection, and evaluation. The first four follow Section V. Segmentation standardises five zone attributes (population, income, footfall, rent, demand) and runs k-means [16], initialised with k-means++ [17], for $k = 2$ to 8. The number of clusters is the value that maximises the silhouette coefficient [18].

A. Huff-Model Evaluation

Every method is scored with the same demand model. The utility that zone i assigns to a store j of attractiveness A_j at distance d_{ij} km is

$$U_{ij} = A_j / d_{ij}^\lambda \quad (3)$$

with distances below 0.5 km set to 0.5 km. For a chosen set S of new stores and the set C of competitors, the share of zone i captured by a new store j is $p_{ij} = U_{ij} / (U_{ij} + \sum_{k \in S} U_{ik} + \sum_{m \in C} U_{im} + u_0)$ (4)

where u_0 is a constant outside option representing online and other channels. Annual revenue and profit of the set are contribution margin and F the fixed annual cost per store.

$$R(S) = \sum_i D_i \sum_{j \in S} p_{ij} \quad (5)$$

$$\Pi(S) = m \cdot R(S) - \sum_{j \in S} (K_j + F) \quad (6) \text{ with } m \text{ the}$$

Three further indicators are reported: the share of demand within $R_c = 2.5$ km of at least one new store (coverage), the demand-weighted mean distance to the nearest new store, and the cannibalisation ratio $\kappa(S) = 1 - R(S) / \sum_{j \in S} R(\{j\})$, which measures how much of the stores' stand-alone revenue disappears because they compete with each other.

B. Selection Methods

Six methods each choose 5 of the 40 candidates. *Population rank* takes the five sites with the most residents within 2.5 km. *Weighted score* ranks sites by a composite of min–max normalised catchment demand d_j , footfall f_j , number of competitors within 2.5 km c_j and annual rent k_j :

$$S_j = w_d d_j + w_f f_j - w_c c_j - w_k k_j \quad (7)$$

The weights are those in Table III, set by the authors and not calibrated. *k-means snap* runs demand-weighted k-means with $k = 5$ on zone coordinates and replaces each centroid by the nearest unused candidate. *Max coverage* solves the maximal covering location problem [11] exactly as a mixedinteger program,

$$\max \sum_i D_i y_i \quad (8)$$

subject to $y_i \leq \sum_{j \in N(i)} x_j$, $\sum_j x_j = p$, and binary x_j , y_i , where $N(i)$ is the set of candidates within R_c of zone i . This model ignores competitors and costs. *Greedy + swap* adds, one at a time, the site that raises Π most, then repeats single-site swaps until none improves Π . *Exhaustive optimum* evaluates Π for every one of the $C(40,5) = 658,008$ five-site sets. A random baseline draws 2,000 random sets.

Because the greedy, swap and exhaustive methods optimise Π under the same Huff model that is used for scoring, they are favoured by construction. The baselines were not tuned to Π . The comparison therefore shows how much is lost when a method optimises something other than profit, not that a Huff-based method would win on real sales.

C. Sensitivity and Monte Carlo Tests

Each selection made at $\lambda = 2$ is re-scored at $\lambda = 1.0, 1.5, 2.0, 2.5$ and 3.0 , and compared with the exhaustive optimum recomputed for each λ . The profit regret of a set is

$$\rho(S) = \frac{(\Pi_{\text{opt}} - \Pi(S))}{\Pi_{\text{opt}}} \times 100 \quad (9)$$

With Π_{opt} the optimum at that λ . The Monte Carlo test draws 500 parameter sets: λ uniform on $[1, 3]$, a zone-level lognormal demand shock ($\sigma = 0.15$) multiplied by a quadrant-level shock ($\sigma = 0.25$), and a lognormal shock ($\sigma = 0.20$) on each competitor's attractiveness. The five-site sets chosen at $\lambda = 2$ are held fixed and re-scored on every draw.

VII. SYSTEM ARCHITECTURE

Fig. 1 shows how the parts connect. The script generates the zone, candidate and competitor tables, cleans them, and writes them to a SQLite database. The same script then runs the SQL queries, the segmentation, the six selection methods and the evaluation, and writes the figures and a JSON file of results. Every number in Section IX is read from that file when the paper is built, so the text and the code cannot drift apart. Runtime dependencies are pandas, NumPy, scikitlearn, SciPy and Matplotlib.

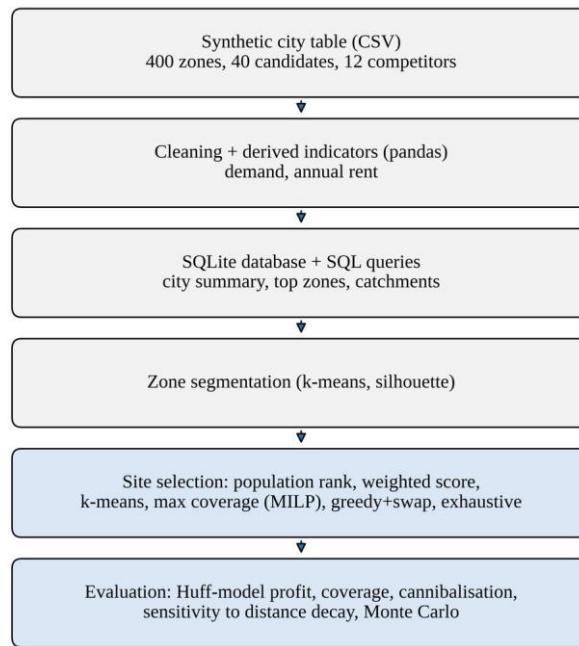


Fig. 1. Architecture of the store-location analytics workflow.

VIII. IMPLEMENTATION

A. Python Implementation

A single script, `store_location_analysis.py`, holds the code. Functions generate the synthetic city, clean it, build the Huff utilities, evaluate any set of sites, run the exhaustive search in vectorised NumPy chunks, and run the greedy-plus-swap search. The mixed-integer program

is solved with `scipy.optimize.milp`. Random seeds are fixed, so a rerun reproduces every table and figure.

B. Database and SQL Analysis

The tables `zones`, `candidates` and `competitors` are written to SQLite with `DataFrame.to_sql`. Three queries are run: city totals (zone count, population, demand, mean income, mean rent), the five zones with the highest demand, and the population and demand inside the 2.5 km catchment of each candidate, using a join on squared distance. The catchment query returned demand between INR 0.61 and 23.46 crore per candidate, and its output matched the pandas calculation exactly. No index or primary key is defined, which is acceptable at this size.

IX. EXPERIMENTAL RESULTS AND ANALYSIS

All figures below come from one run of the pipeline. Profit values are model outputs for the synthetic city, expressed in INR crore per year for the five new stores together.

A. Zone Segmentation

The silhouette coefficient is highest at $k = 2$ (0.561), falling to 0.456 at $k = 3$, with a smaller local maximum of 0.391 at $k = 5$ (Fig. 3). The two-segment solution separates a core of 108 zones from a periphery of 292 zones (Table IV). The core holds 60.0% of demand in 27.0% of the zones, with higher income, footfall and rent. This is what the generator was designed to produce, so it confirms that the clustering recovers built-in structure and says little about real cities.

Table IV: Segment Profile. ($k = 2$)

Segment	Zones	Pop.	Income	Foot.	Rent	Demand %
Core	108	7,299	1.36	62.3	97.3	60.0
Periphery	292	2,437	0.99	26.5	48.5	40.0

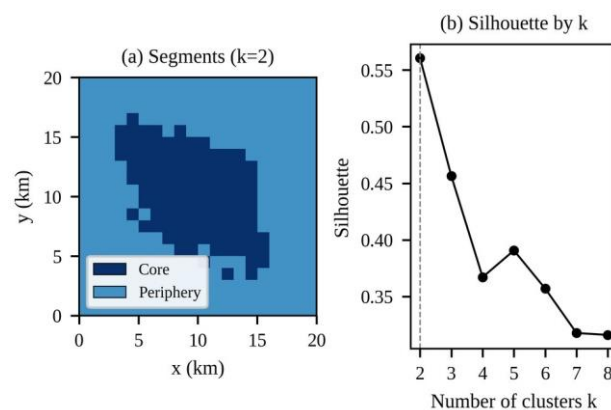


Fig. 3. (a) Zone segments from k-means and (b) silhouette coefficient by number of clusters.

B. Comparison of Selection Methods

Table V and Fig. 4 compare the six methods. The exhaustive search finds a profit of INR 2.67 crore, using sites 16, 18, 30, 32, 36. The four baselines reach 1.83 to 2.31 crore, between 13.5% and 31.6% below the optimum. A random five-site choice averages 1.18 crore (5th–95th percentile 0.28–2.00), and 1.7% of random choices lose money. Across all 658,008 combinations, 98.3% are profitable and only 8 lie within 1% of the optimum.

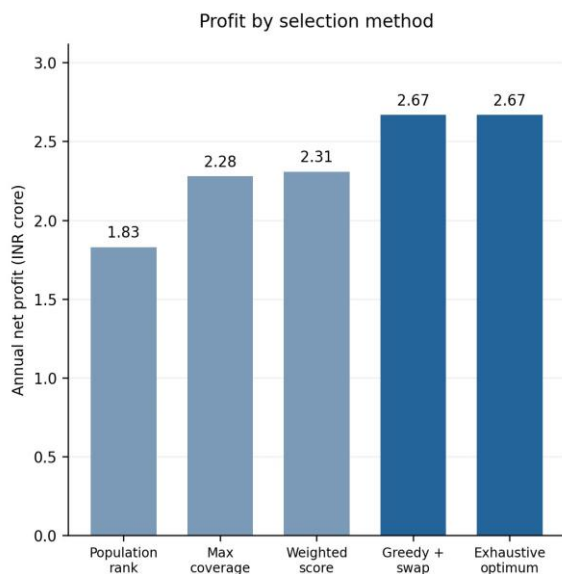


Fig. 4. Reported profits; k-means value not specified.

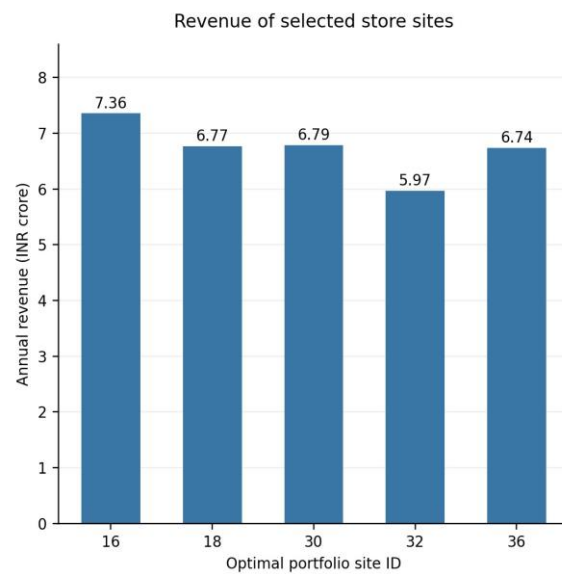


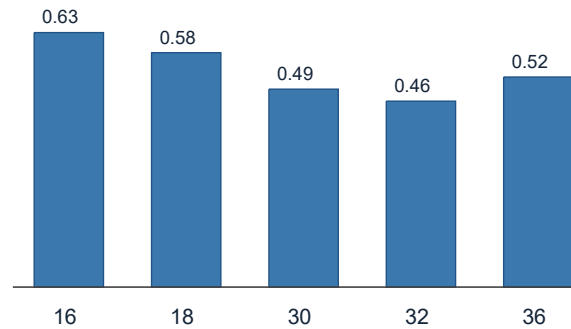
Fig. 5. Revenue by site in the optimal five-store set.

The indicators tell different stories. The covering model reaches the highest coverage (52.2%) but earns 2.28 crore, below the weighted score at 2.31 crore, because it ignores competitors and rent. The k-means snap spreads stores out and has the lowest cannibalisation (10.7%), yet earns less than the score-based choice. The population rank concentrates stores in dense areas, gives the lowest coverage (33.6%) and the longest mean distance (4.17 km), and earns the least of the six. The Huff-based choice accepts more cannibalisation than the two spread-out methods (15.7%) and still earns more, since it places stores where demand outweighs competitor pull. Fig. 5 shows how the optimum and the weighted-score choice sit relative to demand and competitors.

Greedy-plus-swap returns exactly the exhaustive optimum on this instance. The greedy stage alone reaches 2.62 crore, 2.1% below the optimum, so the swap stage matters. On one CPU core the exhaustive search took about 6 s and the heuristic about 0.01 s. Agreement on one synthetic instance does not show that the heuristic is optimal in general, and with more candidates the exhaustive search would grow quickly.

C. Optimal Portfolio

Table VI lists the stores in the optimal set. Store revenue ranges from INR 5.97 to 7.36 crore per year, which.



Optimal portfolio: profit by store

Optimal site ID Net annual profit (INR crore), from Table VI.

corresponds to about INR 29,845 to 36,800 per square foot per year, and every store is profitable on its own. The set captures 23.5% of city demand.

Table VI: Stores in the Optimal Set. (inr crore per year)

Site	Revenue	Rent	Profit
16	7.36	0.25	0.63
18	6.77	0.19	0.58
30	6.79	0.28	0.49
32	5.97	0.17	0.46
36	6.74	0.24	0.52

D. Sensitivity to Distance Decay

Fig. 6 shows regret as λ changes. The optimal set itself changes: at $\lambda = 1.0$ it is sites 5, 16, 18, 24, 36, which shares 3 of five sites with the set chosen at $\lambda = 2$. The Huff-based set stays close to the optimum throughout, with regret of at most 1.1%. The baselines degrade as λ rises. At $\lambda = 3.0$ the regret of the population rank is 90.9%, of the k-means snap 62.8%, of the weighted score 47.1% and of the covering solution 41.0%. Among the baselines, the weighted score has the lowest regret for λ up to 2.0 and the covering solution for λ of 2.5 and above. Holding the Huff-based set fixed is favourable to it, because it was optimised at $\lambda = 2$ and the tests span a range around that value.

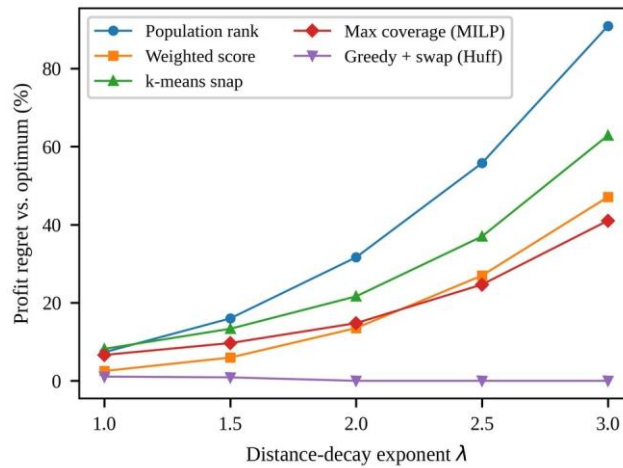


Fig. 6. Profit regret against the optimum for different distance-decay exponents.

E. Monte Carlo Robustness

Table VII summarises 500 random parameter draws. The Huff-based set has the highest mean profit (2.94 crore), the highest 5th percentile (1.05 crore) and no loss-making draws. The weighted score averages 2.60 crore with a loss in 1.0% of draws, and the population rank averages 2.13 crore with a loss in 4.4%. The Huff-based set earned more than the weighted-score set in 97.2% of draws. Spreads are wide (standard deviations of 1.24 to 1.45 crore), and the draws come from the same model family as the evaluation, so this is a test of parameter uncertainty and not of model error.

Table VII: Monte Carlo Profit. (Inr Crore, 500 Draws)

Method	Mean	SD	5th pct.	Loss %
Population rank	2.13	1.45	0.11	4.4
Weighted score	2.60	1.38	0.52	1.0
k-means snap	2.35	1.30	0.40	0.8
Max coverage (MILP)	2.53	1.27	0.61	0.4
Greedy + swap (Huff)	2.94	1.24	1.05	0.0
Exhaustive optimum	2.94	1.24	1.05	0.0

X. DISCUSSION

What was observed. On this synthetic city, the choice of method matters: the spread between the best and worst of the six methods is INR 0.85 crore per year, and the ordering depends on which indicator is read. Coverage, cannibalisation and profit rank the methods differently.

Interpretation. A method that optimises one indicator can look strong on that indicator and weak on another, so the objective has to be chosen before the method. The covering model is the clearest case, since it is exact for its own objective yet not for profit.

Relation to earlier work. The machine-learning studies predict store performance from observed sales [2], [15]. This paper does not: profit here comes from an assumed demand model. What the workflow adds is traceability and a like-for-like comparison, including an exact benchmark that is rarely available for real problems. The two approaches are complementary. A learned performance model could replace the Huff revenue in Eq. (5), and the selection methods here could then be run on top of it.

Practical reading. For an analyst, three habits follow from the results. Report several indicators rather than one. Test selections against a range of plausible distance-decay values, because the ranking of baselines changed within the range tried. Where the candidate list is short enough, compute the exhaustive optimum, since it turns the quality of any heuristic from an opinion into a number.

XI. LIMITATIONS

The main limitation is the data. The city is synthetic, generated from smooth surfaces plus noise, so segment structure and correlations were built in. No claim about any real city follows. Demand, rent and cost parameters were set for plausibility and not estimated. The Huff model has one distance-decay exponent, uses straight-line distance and ignores road networks, store format, parking and brand effects. The methods that optimise the Huff profit are scored under the same model, which favours them. The baseline weights were chosen by the authors and a different choice could reorder the baselines. Only one synthetic city and one random seed were analysed. The sites are all new; no existing own stores are modelled, and the store count is fixed at five. The Monte Carlo test draws from the same model family, and no validation against real sales was possible.

XII. FUTURE WORK

The following items are proposed and have not been implemented. The first priority is to replace the synthetic tables with real data in the same layout: census or ward-level population, observed store sales for calibrating the Huff parameters, mobile-location footfall, and quoted rents. Travel times from a road network should replace straight-line distance. Learned store-performance models such as those in [2] and [15] could supply the revenue estimate. Further steps include several city instances and seeds, existing own stores and capacity limits, multi-year expansion plans, a spatial cross-validation scheme, and an interactive dashboard on top of the SQLite database.

XIII. CONCLUSION

This paper described a compact workflow for store location analysis in which pandas cleans and derives indicators, SQLite and SQL store and summarise the data, kmeans segments the zones, and six selection methods choose five sites that are scored under one Huff-model profit function. On a synthetic city of 400 zones, the Huff-based search earned INR 2.67 crore per year against 1.83 to 2.31 crore for the four baselines, greedy-plus-swap matched the exhaustive optimum, the covering model achieved the highest coverage without the highest profit, and the baselines lost ground as distance decay rose. These values describe the demonstration data and the assumed demand model only.

The practical value lies in the structure: the steps are short, documented and repeatable, and they separate data preparation, storage, selection and evaluation so that each can be checked. Applying the same workflow to real sales and mobility data, and validating the demand model against observed store performance, are the natural next steps.

REFERENCES

1. C. S. Craig, A. Ghosh, and S. McLafferty, "Models of the retail location process: A review," *J. Retailing*, vol. 60, no. 1, pp. 5–36, 1984.
2. J. Lu, X. Zheng, E. Nervino, Y. Li, Z. Xu, and Y. Xu, "Retail store location screening: A machine learning-based approach," *J. Retailing Consum. Serv.*, vol. 77, Art. no. 103620, 2024, doi: 10.1016/j.jretconser.2023.103620.
3. W. McKinney, "Data structures for statistical computing in Python," in *Proc. 9th Python in Science Conf. (SciPy)*, 2010, pp. 56–61, doi: 10.25080/Majora-92bf1922-00a.
4. C. R. Harris *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020, doi: 10.1038/s41586-020-2649-2.
5. F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
6. P. Virtanen *et al.*, "SciPy 1.0: Fundamental algorithms for scientific computing in Python," *Nat. Methods*, vol. 17, pp. 261–272, 2020, doi: 10.1038/s41592-019-0686-2.
7. J. D. Hunter, "Matplotlib: A 2D graphics environment," *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, 2007.
8. H. Hotelling, "Stability in competition," *Econ. J.*, vol. 39, no. 153, pp. 41–57, 1929.
9. D. L. Huff, "Defining and estimating a trading area," *J. Marketing*, vol. 28, no. 3, pp. 34–38, 1964.

10. S. L. Hakimi, "Optimum locations of switching centers and the absolute centers and medians of a graph," *Oper. Res.*, vol. 12, no. 3, pp. 450–459, 1964.
11. R. Church and C. ReVelle, "The maximal covering location problem," *Papers Reg. Sci. Assoc.*, vol. 32, no. 1, pp. 101–118, 1974.
12. C. S. ReVelle and H. A. Eiselt, "Location analysis: A synthesis and survey," *Eur. J. Oper. Res.*, vol. 165, no. 1, pp. 1–19, 2005.
13. N. Roig-Tierno, A. Baviera-Puig, J. Buitrago-Vera, and F. Mas-Verdu, "The retail site location decision process using GIS and the analytical hierarchy process," *Appl. Geogr.*, vol. 40, pp. 191–198, 2013, doi: 10.1016/j.apgeog.2013.03.005.
14. D. Karamshuk, A. Noulas, S. Scellato, V. Nicosia, and C. Mascolo, "Geo-spotting: Mining online location-based services for optimal retail store placement," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Chicago, IL, USA, 2013, pp. 793–801.
15. F. Gao *et al.*, "Location differs between traditional and new retail: A comparison analysis of Starbucks and Luckin Coffee in China using machine learning," *Cities*, vol. 158, Art. no. 105668, 2025.
16. S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.
17. D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proc. 18th Annu. ACM-SIAM Symp. Discrete Algorithms (SODA)*, New Orleans, LA, USA, 2007, pp. 1027–1035.
18. P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, 1987.