HIDDEN DEMAND ANALYSIS

STI Orders

SEPTEMBER 22, 2025

Prabhuraj Krishnamoorthy

Table of Contents

Hidden Demand Analysis

Executive Summary

This project establishes a predictive framework to estimate STI test kit demand across LSOAs (small geographic areas) in the UK.

Using **2019–2025 order data**, enriched with **ONS Census 2021 demographics** and **IMD 2019 deprivation indicators**, a model was designed to:

- Quantify demand at the local level (LSOA).
- Identify **unserved or low-engagement areas** with unmet potential.
- Support **budget allocation and promotional planning** to drive equitable access.

The modelling framework is **fully operational**. Early outputs demonstrate feasibility and provide actionable insights, with accuracy optimisation underway to strengthen reliability.

Key Findings (Data & Business Insights)

Coverage Expansion

Available LSOAs grew from **6,557 in 2019 → 10,997 in 2025 (+68%)**.
Served LSOAs rose from **5,813 → 9,675 (+66%)**.

Unserved Gap

Unserved LSOAs remain 7–13% annually (744 in 2019 → 1,322 in 2025).
Represents missed engagement opportunity despite regional expansion.

Capped vs. Uncapped Regions

Capped regions show higher unserved shares, reflecting constraints in access despite demand.
Uncapped regions sustain broader reach, serving ~88–92% of available LSOAs.

Low-Engagement Areas (≤5 Orders/year)

Persist across years (e.g., 965 in 2024, 1,547 in partial 2025).
Highlight gaps in awareness and accessibility, needing targeted outreach.

Predictive Modelling Readiness

ANN model ingests **80+ census features and 16 IMD deprivation features**. Early results show deprivation, demographics, and housing as **strong predictors**.

Framework already enables **prioritisation of underserved areas**, with calibration underway to refine accuracy.

Business Value

- *Evidence-led planning:* Helps identify where to expand or intensify engagement.
- *Budget optimisation*: Predicts which low-order regions have **high potential**, ensuring promotional resources are not wasted.
- *Equity focus*: Highlights structural inequalities (capped vs. uncapped), guiding fairer service delivery.

Exploratory Data Analysis

Region-wise Trends (2019–2025)

The data tracks LSOA coverage between 2019 and 2025, showing how many areas were available, how many were served, and where gaps remain.

1: Active regions and LSOA availability per year:

year	Total regions	Total lsoas available	Capped regions	capped lsoas_available	Uncapped regions	Uncapped lsoas_available
2019	27	6730	9	2004	18	4726
2020	33	8028	10	2153	23	5875
2021	38	8718	11	2188	27	6530
2022	43	9536	14	2638	29	6898
2023	42	9534	15	2875	27	6659
2024	45	9275	17	3466	28	5809
2025	53	10997	29	5781	24	5216

Business Insights

- *Market Expansion:* LSOA coverage expanded steadily (+63% from 2019–2025).
- *Capping Effect:* The share of capped regions grew over time, limiting flexibility in service

Yearly LSOA Coverage (2019–2025)

The data tracks coverage at the LSOA level between 2019 and 2025, showing how many areas were **available**, how many were **served**, and where gaps in service remain.

Table 2: Served Vs Unserved LSAO per year

Year	Available LSOAs	Served LSOAs	Unserved LSOAs
2019	6557	5813	744

2020	8028	7379	649
2021	8718	7942	776
2022	9536	8288	1248
2023	9534	8366	1168
2024	9275	7997	1278
2025	10997	9675	1322

Figure 1: Served Vs Unserved LSAO per year

Insights:

- *Market Expansion*: Coverage widened from **6,557 → 10,997 LSOAs** (+68%).
- *Service Growth*: Served areas increased in parallel (**+66% growth**).
- *Stable Efficiency*: Served share remained steady at **~88–90%**, indicating expansion with less efficiency gain.
- *Persistent Gap*: **744 → 1,322 unserved LSOAs** each year (~7–12%).
- *Business Opportunity*: Growth now lies in **closing unserved gaps**, along with adding new regions.

Capped vs Uncapped Regions: LSOA Coverage (2019–2025)

This data compares the number of available, served, and unserved LSOAs in **capped vs uncapped regions** over time. It highlights how policy settings influence service reach. *Table 3: Lsoa coverage on Capped Vs. Uncapped*

year	Uncapped – Available LSOA	Capped- Available LSOA	Uncapped served LSOA	Capped - served LSOA	Uncapped – Missed LSOA	Capped - Unserved LSOA
2019	4726	1831	4423	1390	303	441
2020	5875	2153	5492	1887	383	266
2021	6530	2188	5856	2086	674	102
2022	6898	2638	5871	2417	1027	221
2023	6659	2875	5944	2422	715	453
2024	5809	3466	5147	2850	662	616
2025	5216	5781	4610	5065	606	716

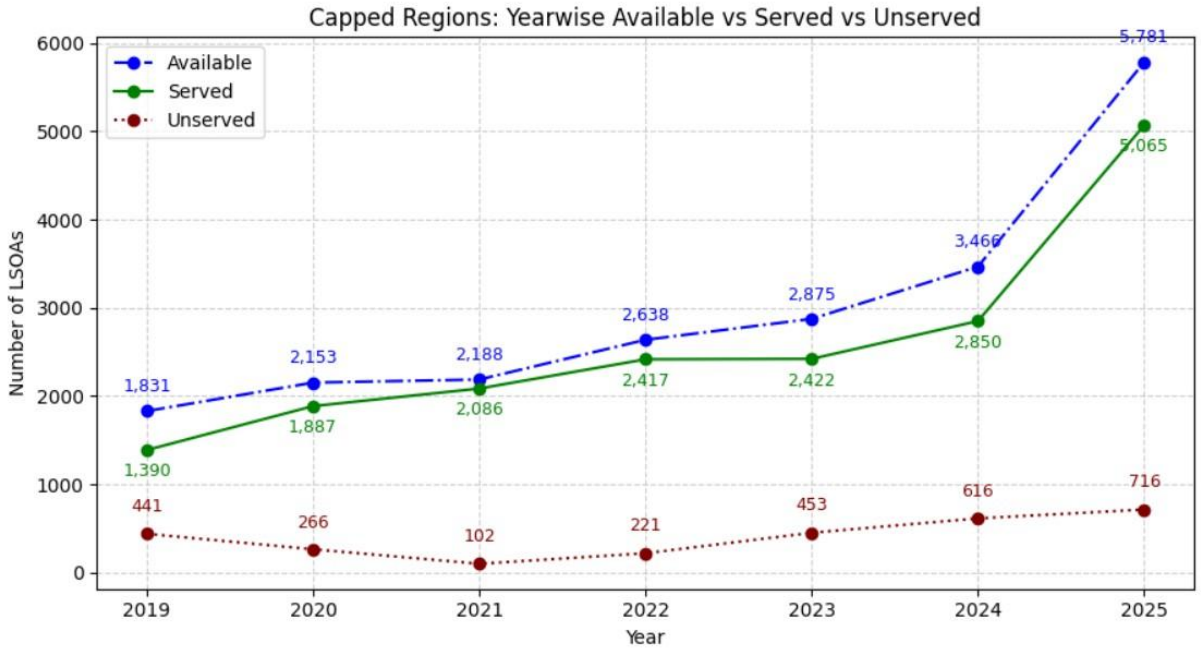


Figure 2: Unserved LSAO's per Year on Capped regions

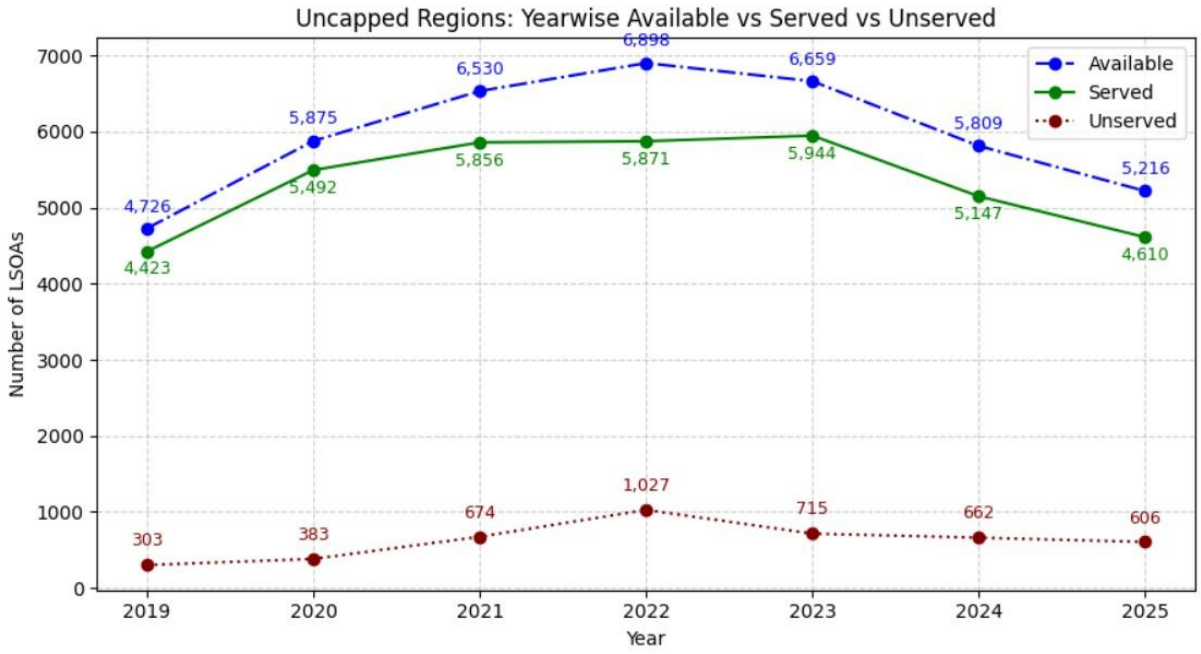


Figure 3: Unserved LSAO's per Year on Uncapped regions

Business Insights:

- *Uncapped regions* consistently serve the majority of their available LSOAs, maintaining broad coverage.
- *Capped regions* show a higher share of unserved LSOAs, suggesting that capping constrains access despite demand.
- The **gap between capped and uncapped coverage persists year-on-year**, pointing to structural inequalities in reach.
- In 2025, uncapped regions served ~88% of available LSOAs, while capped regions served ~88% as well due to movement of Uncapped regions to capped.

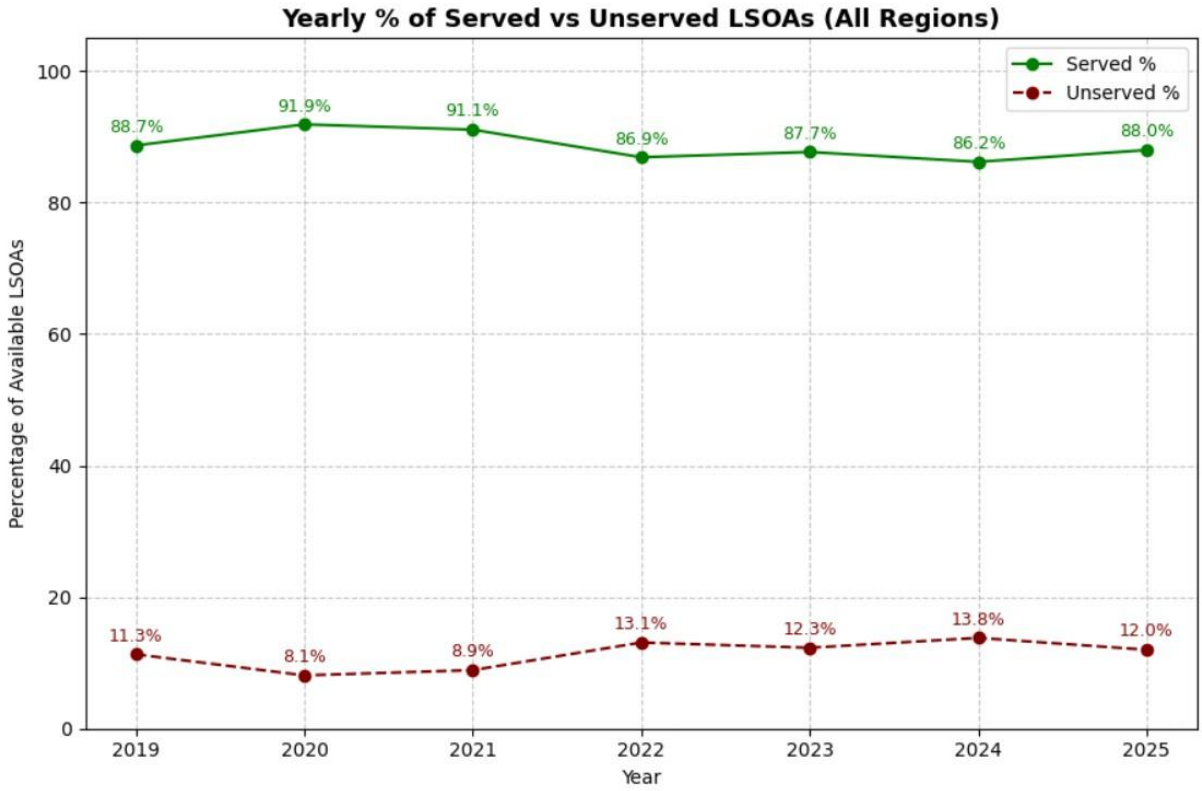


Figure 4: Percentage over served & Unserved regions

year	Available LSOAs	Served LSOAs	Unserved LSOAs	Served %	Unserved %
2019	6557	5813	744	88.7	11.3
2020	8028	7379	649	91.9	8.1
2021	8718	7942	776	91.1	8.9
2022	9536	8288	1248	86.9	13.1
2023	9534	8366	1168	87.7	12.3
2024	9275	7997	1278	86.2	13.8
2025	10997	9675	1322	88	12

Low-Engagement Areas (≤5 Orders per LSOA)

About:

This section tracks the number of LSOAs each year with **very low order volumes (≤5 orders)**. These represent the areas where engagement and uptake are weakest.

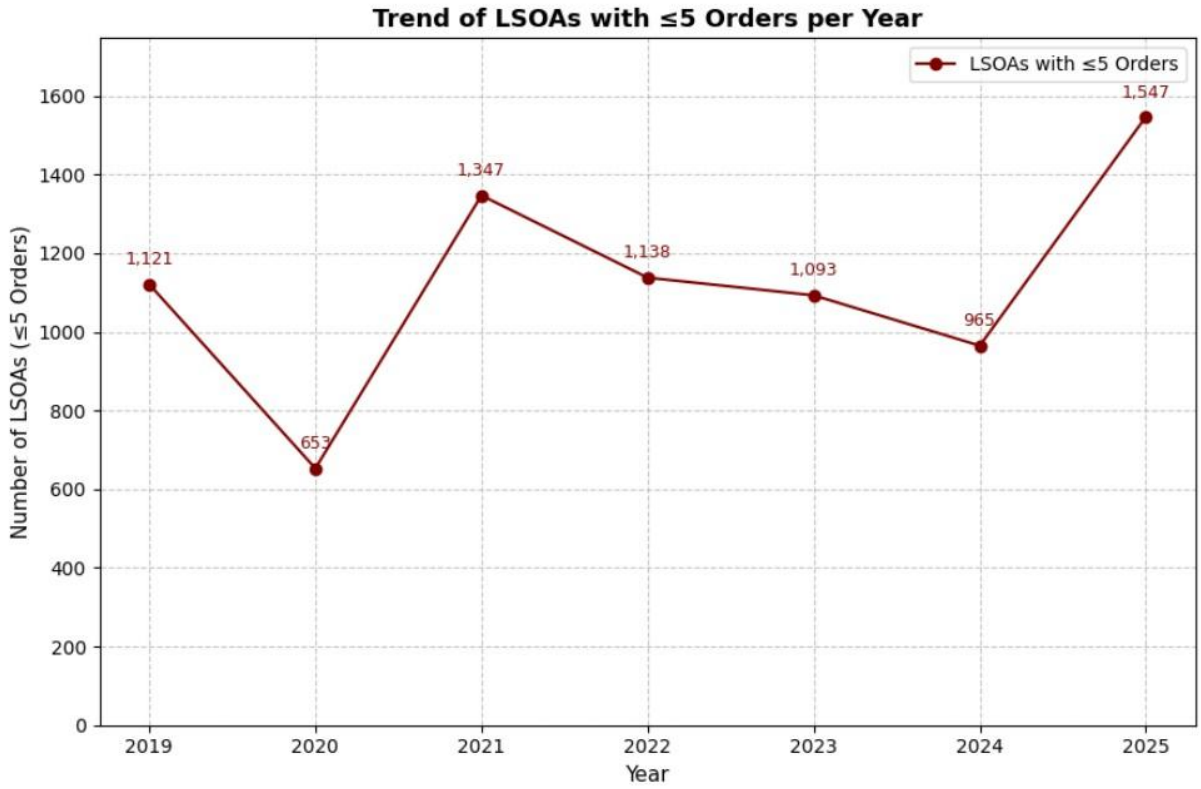


Figure 5: LSOA count with <= 5 Orders per year

2025 is incomplete**, so the spike (1,547) should be read with caution.**

Key Insights

- Awareness Gaps: Persistent clusters of low-order LSOAs point to areas where awareness or access is limited.
- Targeted Outreach: These regions may benefit most from promotional campaigns and community-level engagement.
- Opportunity Indicator: Tracking low-engagement areas helps flag unrealised demand and potential for growth.

Predictive Modelling for LSOA Potential

Objective:

The model estimates the **true potential demand** for each LSOA by learning from demographics, deprivation, and historical order data. This helps to:

- Identify **unserved areas** with high unmet need.
- Flag **low-order LSOAs** where uptake is below potential.
- Guide **promotion and awareness budgets** for sustainable engagement.

Input Data

- ONS Census 2021: 66 tables sourced, 16 pre-processed. ~130 raw features engineered into ~80 derived features, covering demography, health, housing, education, ethnicity, and migration.
- IMD 2019: 12–16 socio-economic deprivation indicators across income, employment, health, crime, housing, and access barriers.
- Orders (2019–2023): Annual LSOA-level snapshots created from order history.

Example Transformation (Housing – Accommodation Type)

Original Census Columns (Census 2021 extract):

- Accommodation type: Detached
- Accommodation type: Semi-detached
- Accommodation type: Terraced
- Accommodation type: Flats (purpose-built, converted, commercial building)
- Accommodation type: Temporary (e.g., caravan, mobile structure)

Transformation Process:

- Columns renamed for clarity.
- Categories grouped into **density types**:
    - *Low density* → Detached + Semi-detached
    - *Mid density* → Terraced
    - *High density* → Flats & converted/shared housing
    - *Temporary housing* → Caravans/mobile structures
- Converted into **percentages of total households**, enabling fair comparison across LSOAs of different sizes.

date	geography	geography code	Accommodation type: Total: All households	Accommodation type: Detached	Accommodation type: Semidetached	Accommodation type: Terraced	Accommodation type: In a purpose-built block of flats or tenement	Accommodation type: Part of a converted or shared house, including bedsits	Accommodation type: Part of another converted building, for example, former school, church or	Accommodation type: In a commercial building, for example, in an office building, hotel or over a shop	Accommodation type: A caravan or other mobile or temporary structure
2021	City of London	E01000001	837	0	3	13	803	0	11	7	0
2021	City of London	E01000002	825	1	2	29	769	2	13	9	0
2021	City of London	E01000003	1017	1	0	0	994	2	13	7	0
2021	City of London	E01000005	479	0	0	2	457	5	5	10	0

Figure 6: Data sample for Accommodation type

```python
def preprocess_accommodation_type(df):
    # Rename columns for clarity
    df = df.rename(columns={
        "Accommodation type: Total: All households": "total_households",
        "Accommodation type: Detached": "detached",
        "Accommodation type: Semi-detached": "semi_detached",
        "Accommodation type: Terraced": "terraced",
        "Accommodation type: In a purpose-built block of flats or tenement": "flats_purpose_built",
        "Accommodation type: Part of a converted or shared house, including bedsits": "converted_shared_house",
        "Accommodation type: Part of another converted building, for example, former school, church or warehouse": "converte
        "Accommodation type: In a commercial building, for example, in an office building, hotel or over a shop": "commercia
        "Accommodation type: A caravan or other mobile or temporary structure": "caravan_or_temp"
    })

    # Group similar categories and calculate percentages
    df["pct_low_density"] = (df["detached"] + df["semi_detached"]) / df["total_households"] * 100
    df["pct_mid_density"] = df["terraced"] / df["total_households"] * 100
    df["pct_high_density"] = (
        df["flats_purpose_built"] +
        df["converted_shared_house"] +
        df["converted_other"] +
        df["commercial_building"]
    ) / df["total_households"] * 100
    df["pct_temp_housing"] = df["caravan_or_temp"] / df["total_households"] * 100

    # Final output with only useful, shrunk features
    result = df[[
        "geography", "geography code",
        "pct_low_density", "pct_mid_density", "pct_high_density", "pct_temp_housing"
    ]].copy()
    save_file(result, "accomodation.csv")

preprocess_accommodation_type(open_file("census2021-ts044-lsoa.csv"))
```

Figure 7: code sample used to implement Conversion

Transformed Table (LSOA-level % features)

Table 4: Transformed + Normalized features

lsoa_code	pct_low_density	pct_mid_density	pct_high_density	pct_temp_housing
E01000001	0.358422939	1.553166069	98.08841099	0
E01000002	0.363636364	3.515151515	96.12121212	0
E01000003	0.098328417	0	99.90167158	0
E01000005	0	0.417536534	99.58246347	0
E01000006	22.02166065	51.08303249	26.71480144	0.180505415

Summary (ONS Census Features)

- 16 most relevant tables were processed.
- All features scaled to percentage levels for comparability across LSOAs.
- Final dataset: ~80 derived features ready for modelling.

Index of Multiple Deprivation (IMD 2019)

Raw Data:

40+ columns per LSOA, including score, rank, and decile for each domain.

Processing:

- Retained only score columns (rates/percentages).
- Dropped ranks and deciles to reduce redundancy.
- Final output: 12–16 domain features per LSOA, directly usable for modelling.

Example of Simplification:

Original columns (Employment & Education): ○ Employment Score (rate) ○ Employment Rank (where 1 = most deprived) ○ Employment Decile (where 1 = most deprived 10%)

○ Education, Skills and Training Score ○ Education, Skills and Training Rank ○ Education, Skills and Training Decile

Converted to:

○ employment_score
○ education_score

Final IMD Feature Columns (per LSOA)

Each LSOA has one **percentage score per domain**:

Feature Name	Description
`imd_score`	Overall IMD index score
`income_score`	Income deprivation score
`employment_score`	Employment deprivation score
`education_score`	Education, skills & training score
`health_score`	Health deprivation & disability score
`crime_score`	Crime score
`housing_barrier_score`	Barriers to housing & services (combined score)
`living_env_score`	Living environment score `idaci_score` Income deprivation affecting children index `idaopi_score` Income deprivation affecting older people index `child_young_ppl_score` Children & young people subdomain `adult_skills_score` Adult skills subdomain

Feature Name	Description
`geo_barriers_score`	Geographical barriers subdomain
`wider_barriers_score`	Wider housing barriers subdomain `indoors_score`

Indoors living environment subdomain `outdoors_score` Outdoors living environment subdomain

Summary

- **Ranks and deciles** were removed to avoid redundancy.
- **Scores only** retained — interpretable as rates or percentages.
- Final dataset: **16 simplified features** covering deprivation across income, employment, education, health, crime, housing, environment, and access barriers.

Order Data (2019–2023)

Base Orders Extraction

- *Timeframe:* Orders from 2019–2025.
- *Geography:* United Kingdom only, with valid lsoa_code or lsoa_name.
- *Product:* STI Test kits (brand = 1).
- *Contracts:* Orders retained only if they fall within an active billing contract for that region, or if the region mapping is missing (unmapped).

Region Mapping

- Orders mapped to regions using lsoa_prefix via lsoa_to_sh_24_region.
- If no match found, fallback to direct region name matching.
- Result: each order assigned to a region wherever possible.

Contract Validation

- Orders included if:
    - Order date falls within contract start and end dates, OR
    - No region mapping is available (sh_region IS NULL).

Final Order Selection & Aggregation

- Excluded test/fake regions: *PrEP Trial%*, *Northern Ireland%*, *Freetesting%*, *Test Region%*.
- Aggregated by year, month, LSOA, region.
- Flagged whether the region was capped or uncapped (region_quotas).

- Counted distinct orders (COUNT(DISTINCT xx_uid)).
- Retained only rows with a non-empty LSOA code.

Preparation Steps

Handling Missing LSOA Names

- Built a (year + lsoa_code → lsoa_name) mapping from rows with valid names.
- Used the mapping to fill missing names only for null rows.
- Dropped any remaining null values.
- Excluded the current month to avoid incomplete data bias.

Building the Global Reference List of LSOAs

Objective: Create a full reference of all UK LSOAs per region to benchmark against the orders data.

Steps Taken:

1. Global LSOA Collection: Gathered the complete UK list (lsoa_code, lsoa_name).
2. Prefix Extraction: Derived an lsoa_prefix column from lsoa_name for standardised mapping.
3. Region Mapping: Applied the lsoa_reg_map lookup to assign each LSOA to its region (final_sh_region).
4. Region-wise Availability: For each region, calculated:

   Count of all available LSOAs.

   List of LSOAs in that region.

5. Comparison with Orders Data → Matched the global list against served LSOAs from orders:

   Served LSOAs = present in both orders and global list.

   Unserved LSOAs = present in global list but missing in orders.

Data Cleaning & Preprocessing

Orders Selection

- Filtered orders between **2019–2023**.
- Current working dataset: **orders_2019_23**.

Alignment with External Data

- **ONS** (Census 2021) and **IMD** (Deprivation 2019) values are reported **yearly**.
- To make orders comparable, we created an **annual snapshot**.

Annual Snapshot Creation

- Applied a **running average** to smooth monthly & yearly fluctuations.
- For each **LSOA**, derived an **order snapshot per year** across the 5 years (2019– 2023).

Merging ONS and IMD Tables

ONS Data Merge

- Multiple **ONS tables** (Census 2021) were merged with the **orders dataset** using lsoa_code.
- Function logic: loop through files → read table → join on lsoa_code.

Observation:
- Orders table contains **more LSOAs** than ONS tables.
- Missing values (NaN) appear where **new LSOA codes** exist in orders but not in older ONS data.

IMD Data Merge

- IMD (2019) dataset merged with orders on lsoa_code.

Outlier Detection & Filtering

Observation

- Order data is **heavily right-skewed**, with many LSOAs having just **1–10 orders per year**.

Problem

- If included directly, the model may overfit to **low-order cases**, reducing accuracy for regions with higher true demand.

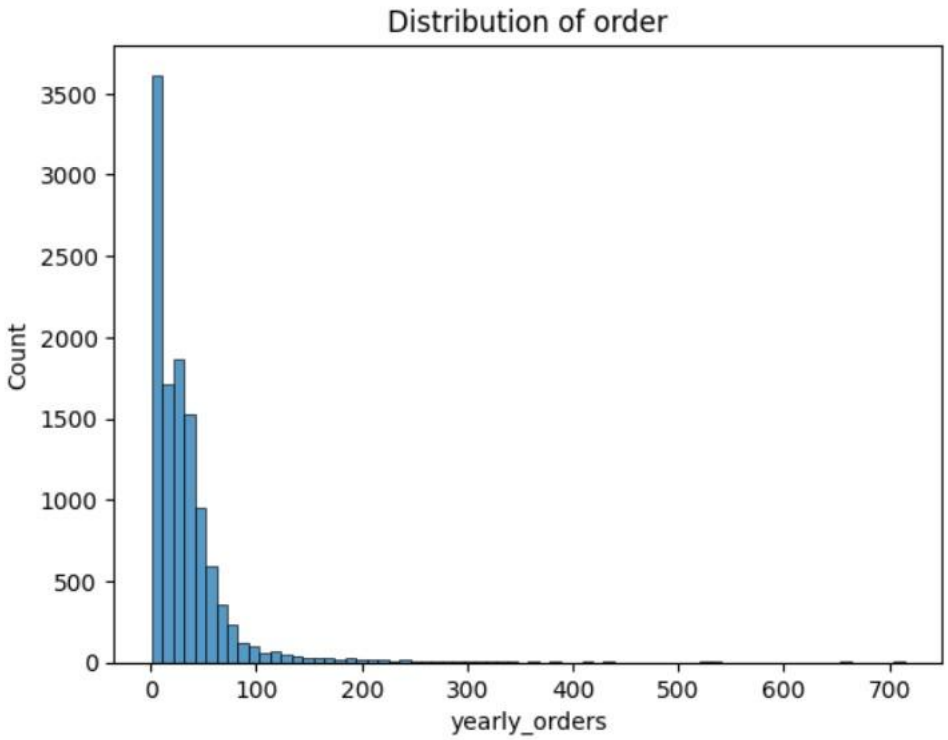


Figure 8: Distribution LSOA vs orders count

Action

- Applied **box plots** and the **Interquartile Range (IQR)** method to detect unusually low or high order counts.

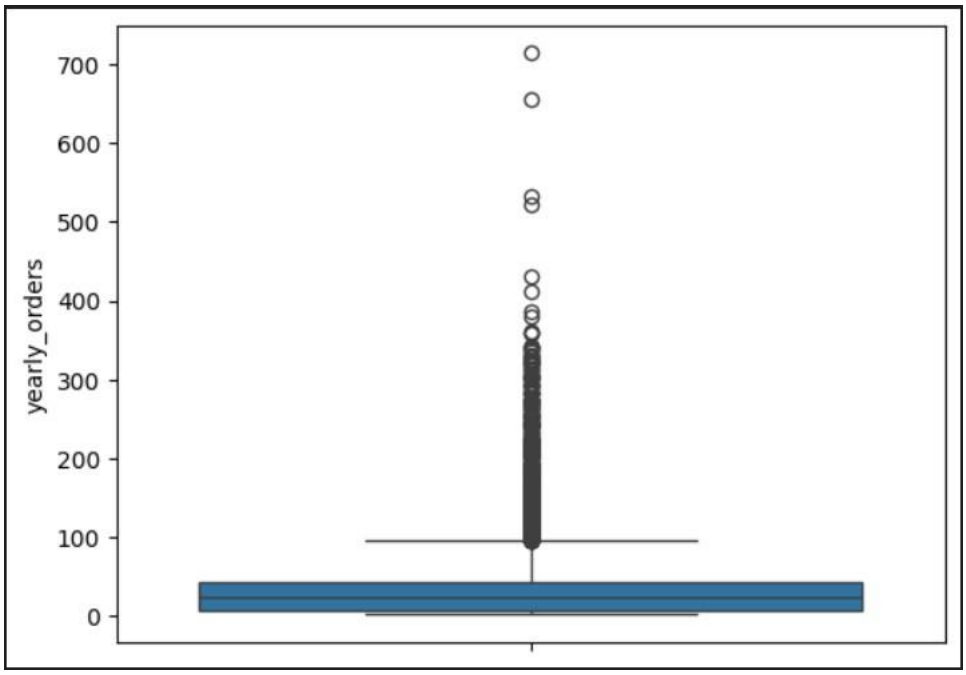


Figure 9: Box plot on outlier distribution

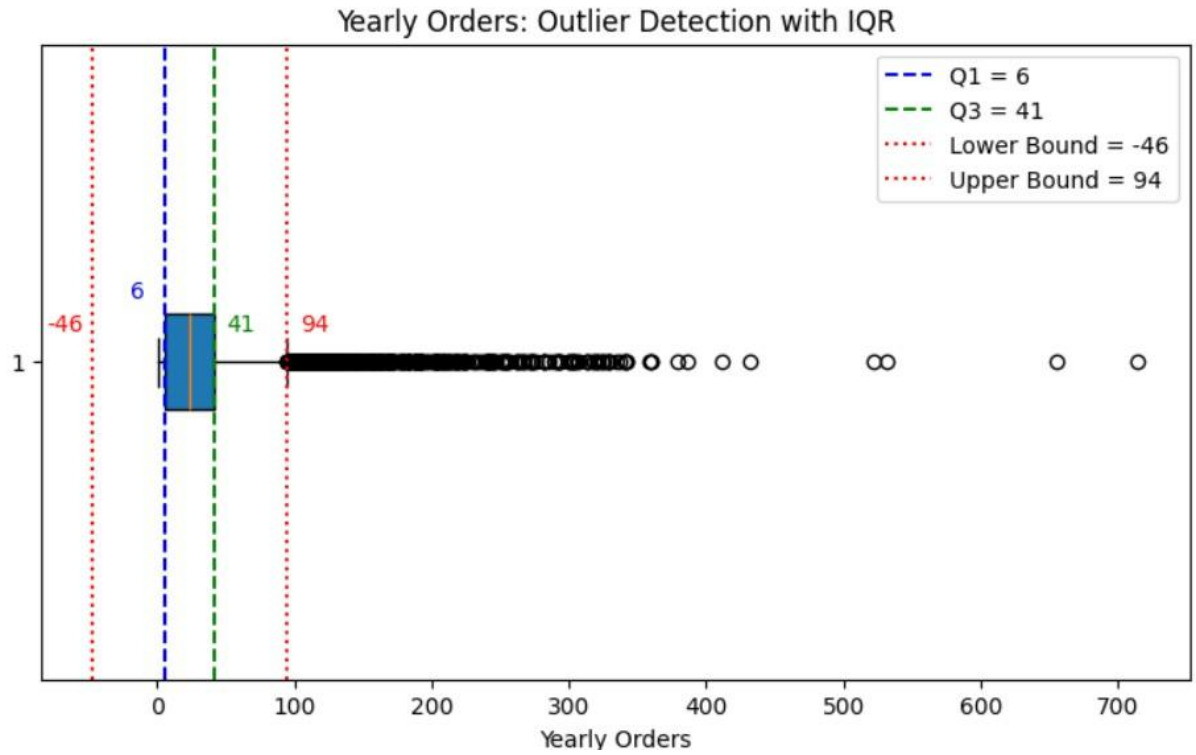


Figure 10: IQR plot describing the boundaries

Outlier Filtering as a Hyperparameter

- Outlier thresholds (lower/upper bounds) were treated as a hyperparameter to be tuned.
- Aim: Balance accuracy improvement with fairness in keeping enough LSOAs in the dataset

Understanding Q1, Q2, Q3 and IQR

- *Q1 (First Quartile):* **6** → 25% of the data falls below this point. It marks the lower edge of the "typical" range.
- *Q2 (Median):* **24** → The midpoint: half the LSOAs have ≤ 24 orders, half have more.
- *Q3 (Third Quartile):* **41** → 75% of the data falls below this point. It marks the upper edge of the "typical" range.
- *IQR (Q3 − Q1):* **35** → Spread of the central 50% of the data — the stable "core".

From these values:

- *Lower Bound*: **-46** (not meaningful, since orders can't be negative).
- *Upper Bound*: **94** → Values above this may be treated as outliers.

Business Decision

- A large share of LSOAs cluster at very low order counts (**1–10 orders per year**).
- Including these directly reduces accuracy, as the model tends to **predict low everywhere**.
- If two LSOAs share similar characteristics but differ due to **temporary factors** (e.g., promotions), the model may get confused in assigning prediction weights.

Therefore, testing different thresholds on order counts is essential — this acts as a hyperparameter to balance accuracy vs inclusivity.

Plan to fine tune this Hyperparameter:

1. **Choose thresholds** (e.g., ≤0, ≤3, ≤5, ≤10, ≤15).
2. **Filter training data** below each threshold (test data left untouched).
3. **Train and evaluate** the model (R^2, MAE, RMSE).
4. **Compare results** across thresholds ❼ see accuracy vs. data coverage tradeoff.
5. **Select best threshold (N)** = balance of improved accuracy and keeping enough LSOAs for fairness.

Note: Threshold N is treated as a **hyperparameter**.

Correlation Analysis to Filter Features

Purpose

Reduce noise by removing irrelevant or redundant features.

Retain only the most meaningful predictors for modelling orders.

Step 1: Target Correlation

- Compared each numeric feature with the **target (orders)**.
- Dropped weak features using a **correlation threshold = 0.2**.
- Retained **40 features out of 92**.

  ❼ *Hyperparameter:* the **threshold value (0.2)** can be tuned higher/lower depending on model performance.

Step 2: Feature-to-Feature Correlation

- Checked for **strongly correlated feature pairs** to avoid redundancy.
- Filtered pairs with **|correlation| ≥ 0.9**.

  ❼ *Hyperparameter***:** the **cutoff (0.9)** for high correlation.

Examples:

1. pct_youth_15_24 ↔ pct_students (0.986) → Highly overlapping demographic measure.

2. pct_religious ↔ pct_no_religion (-0.984) → Opposite measures of the same concept.

3. imd_score ↔ income_score (0.968) →
   - *Note*: IMD is a *composite index* built from domains like income, health, education, crime, etc.
   - Even with overlap, individual scores (like income or health) hold **independent explanatory power** for orders.
   - In this case, we focus on **correlation of individual score with the output (orders)** rather than holding imd_score and removing individual scores.

Final Output

A refined set of features, representing **predictors of order values**.

These predictors allow us to **explain and interpret order patterns** in terms of demography, deprivation, and housing context.

If accuracy is unsatisfactory, thresholds for correlation filtering (**0.2 for target, 0.9 for features**) can be revisited — treated as **tunable hyperparameters**.

Feature Importance Validation (Random Forest)

Approach

- A **Random Forest (RF)** model was applied to rank features by their contribution to predicting order volumes.
- The top contributors were compared with correlation analysis results to validate consistency.
- A **heatmap** was used to visualise feature contributions, highlighting their intensity and relative importance.

Key Feature Groups Identified

Socio-economic deprivation deprivation_severity_index,
income_score, idaci_score,
deprived_3d_pct.

Demographics & housing pct_youth_15_24,
pct_young_adult_25_34,
pct_older_50_plus, pct_non_uk_lt_2yr,
pct_black, pct_mid_density,
pct_under_occupied, small_2to3_hh_pct,
population_density_Persons_per_sq.km.

Labour & unemployment pct_recently_unemployed,
pct_long_term_unemployed.

Health & wellbeing health_score,
child_young_ppl_score,
crime_score,
wider_barriers_score,
education_score.

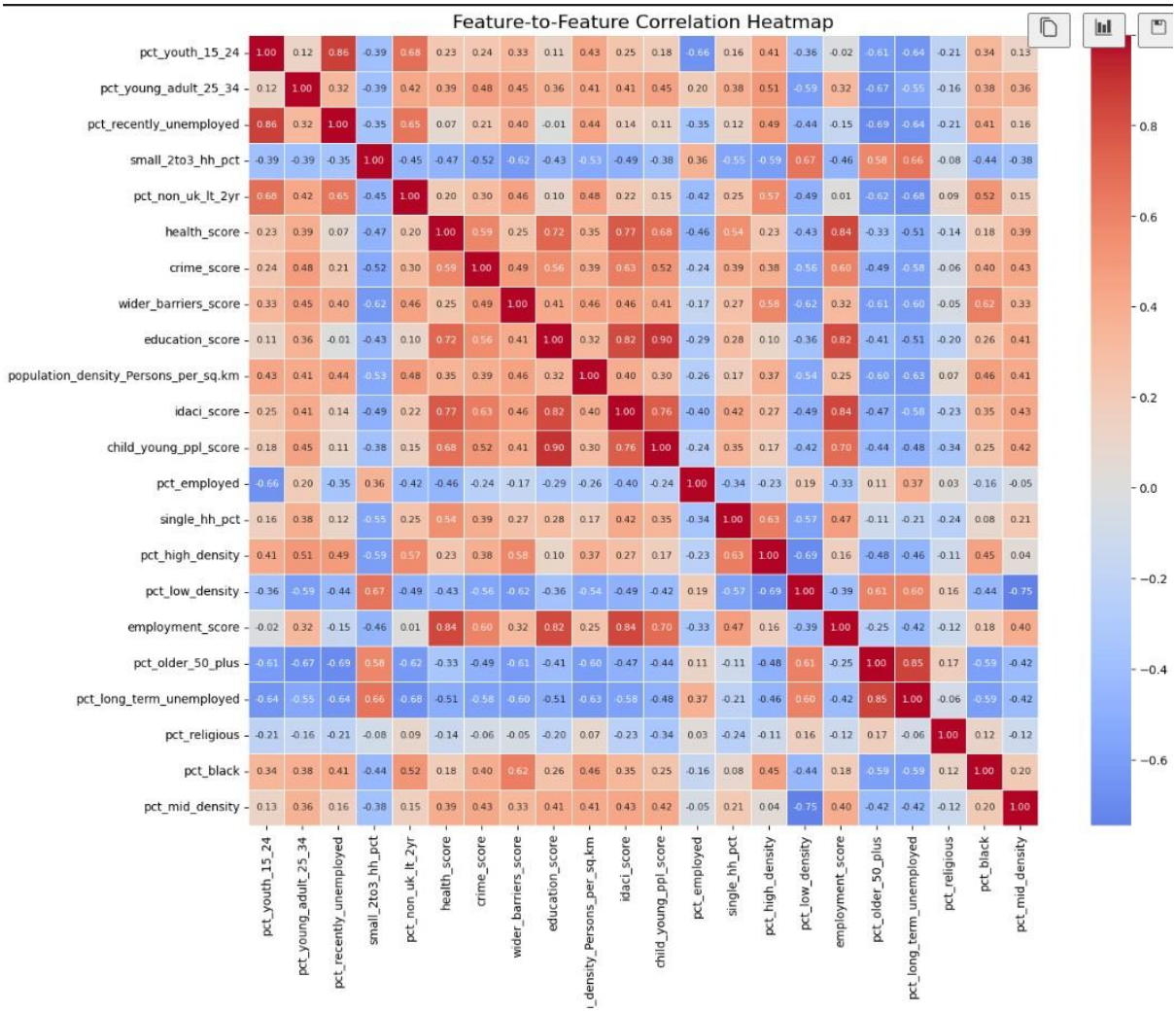
Cultural pct_religious.

Insights

Consistency: RF confirms that deprivation, demographics, and health indicators are the strongest drivers of order volumes.

Overlap: Some predictors show moderate redundancy (e.g., health_score vs employment_score, correlation ≈ 0.84). Trade-offs are required when selecting features.

Refinement: Final feature selection may still need tuning — **keeping or dropping specific features** could improve model accuracy.



Predictive Model: Artificial Neural Network (ANN)

The predictive model was first built in the earlier MVP framework and is now being **refined with additional hyperparameters and filtered data**. This iterative approach is aimed at improving accuracy and producing reliable forecasts. While refinement is ongoing, the key steps and their purpose are outlined below

Actions Taken

Data Preparation

- Cleaned missing values and aligned order data with ONS & IMD datasets. Scaled all features to ensure fair comparison across LSOAs.

Model Design (ANN)

- *Input layer:* Each feature (e.g., demographics, deprivation, housing) is passed into its own node, ensuring every signal enters the model.
- *Hidden layers:* Three layers (128, 64, and 32 nodes) combine and transform inputs, uncovering patterns that are not directly visible.

- *Output layer:* Produces a single numeric value — the predicted number of orders per LSOA.

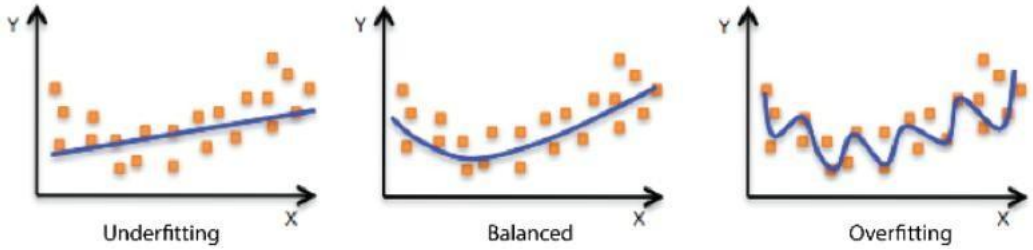
Model Safeguards: Dropout & Early Stopping

Dropout (20%)

What it does (technical): Randomly ignores 20% of the nodes during training to prevent the model from memorising exact patterns.

Why it matters (business):

- Without dropout, the model might "remember" that LSOA1 had exactly 100 orders, instead of learning *why* it had 100.
- With dropout, the model learns the underlying drivers (e.g., demographics, deprivation), so predictions remain reliable for unseen LSOAs.



Early Stopping

What it does (technical): When training, the model keeps adjusting itself to fit past data. If we let it go too long, it may start overfitting — learning noise and quirks. This step Stops training when improvements on unseen validation data level off and restores the best-performing version of the model.

Why it matters (business):

- Prevents "overfitting," where the model learns irrelevant noise from past data.
- Ensures predictions generalise to real-world regions, not just the training set.

Train-Test Split & Scaling

Action:

- Split data into train and test,
- fit the scaler only on the training set and - transform the test set.

Benefit:

Prevents information leakage - if we scale using the whole dataset, the mean, median, and standard deviation of the test set would "leak" into training, giving an unfair advantage and misleading accuracy.

Reasons for These Choices

- **Neural networks** capture complex relationships between demographics, deprivation, and orders.
- **Scaling & careful splitting** prevents bias or data leakage.
- **Dropout + early stopping** build a more reliable and generalisable model.

Results

The model tested on unseen data reported: with below metrics, ○ **R^2:** how much of the variation in orders the model can explain ○ **MAE** (Mean Absolute Error): the average difference between predicted and actual orders.

Example: If MAE = 30, on average the model is off by about 30 orders per LSOA.

○ **RMSE** (Root Mean Squared Error): shows the typical size of errors, with bigger mistakes penalised more.

Example: If RMSE = 50, most predictions are within ±50 orders, but it highlights when a few LSOAs are very far off.

SHAP Analysis

- *Why*: Helps us understand **which features drive the model's predictions** (e.g., deprivation, demographics, housing).
- *Benefit*: Builds trust — instead of a "black box," we can show *why* the model predicts high or low orders for a given LSOA.

Training vs Validation Loss

- *Why*: Tracks how well the model learns during training and whether it generalises.
- *Benefit*: Allows us to adjust hyperparameters (epochs, dropout, learning rate) if the model shows signs of overfitting or underfitting.

Steps enhancing accuracy:

- *Feature refinement*: add/remove socio-economic, demographic, and health features.
- *Hyperparameter tuning*: Adjust learning rate, batch size, dropout rates, and thresholds to optimise model performance.
- *Model comparison*: Benchmark against Random Forest or Gradient Boosted Trees to test whether simpler or ensemble models provide stronger or more interpretable results.