Task 1: Solving an Operational Problem (NLP & Optimization) Report

0. Data Loading and Exploration

Objective:

The goal of this step is to load the delivery dataset and perform initial exploration to understand its structure, contents, and missing values. This step is essential to prepare the data for subsequent address extraction, location estimation, and courier assignment tasks.

Methodology:

1. **Importing Libraries:**
    a. `pandas` and `numpy` for data manipulation.
    b. `re` for regular expressions (used later for address parsing).
    c. `sklearn` libraries for text vectorization and similarity computation.
    d. `matplotlib` and `seaborn` for visualization.
    e. `scipy.spatial.ConvexHull` for calculating operational areas.
2. **Loading Data:**
    a. The dataset `DeliveriesDataRetracted.csv` was loaded using `pd.read_csv()`.
    b. The first few rows were inspected using `df.head()`.
    c. Column names, data types, and non-null counts were checked using `df.info()`.
3. **Initial Observations:**
    a. Dataset contains **6 columns**: `DeliveryAddressFirstLine`, `DeliveryAddressSecondLine`, `DeliveryDate`, `DeliveryLat`, `DeliveryLon`, and possibly an index column.

b. Delivery dates range from **2025-10-11 to 2025-10-16**.
c. Data for **15-10-2025** contains 308 deliveries, while **16-10-2025** has 313 deliveries.
d. For **16-10-2025**, latitude and longitude are missing for all entries, which will require estimation later.

4. **Missing Values:**
   a. Checked using `df.isnull().sum()`.
   b. Confirmed that only **DeliveryLat** and **DeliveryLon** for 16-10-2025 are missing, while other fields are mostly complete.

1. Extracting Meaningful Address Fields

Objective:

The goal of this step is to transform unstructured and noisy Arabic delivery addresses into structured fields. This allows downstream processes such as geolocation estimation and courier route optimization. Key fields include **Building Number, Street Name, District, Area Details, Floor, Apartment, and Landmark**.

Methodology:

1. **Data Cleaning:**
   a. Converted Arabic letters to standard forms (e.g., أ, إ, آ → ا, ى → ي, ة → ه).
   b. Converted Arabic numerals to English (٠١٢٣٤٥٦٧٨٩ → 0123456789).
   c. Removed special characters and extra spaces.
   d. Dropped the `DeliveryAddressSecondLine` column, focusing on the first line for address extraction.
2. **Field Extraction:**
   a. **Building Number:** Detected patterns like "عماره 14", "برج 12", "مبني 4", "بلوك 7" or addresses starting with a number.

b. **Street Name:** Extracted using keywords "شارع" or "ش" and refined using a stopwords list. Known street names were also matched explicitly.
c. **District:** Since all addresses belong to Nasr City, the district was set as "مدينه نصر".
d. **Area Details:** Extracted using patterns for neighborhoods or regions, e.g., "المنطقه الاولى", "8 الحي".
e. **Floor:** Extracted floor numbers or descriptors (e.g., "بدروم", "ارضي", "الدور الثاني").
f. **Apartment:** Detected patterns like "شقه 3" or "apartment 3".
g. **Landmark:** Identified nearby references using keywords like "امام", "خلف", "بجوار", "قريب من" and cleaned extra information.

3. **Improved Extraction (v2):**
   a. Refined regex patterns to catch additional variations.
   b. Applied hierarchical extraction logic to reduce missing values.
   c. Street and building numbers were further standardized and verified against known streets.

Results:

- Missing Value Percentages After Extraction (v2):

Field	Missing Percentage
BuildingNumber_v2	41.86%
StreetFinal	20.93%
AreaDetails_v2	90.75%
Floor_v2	69.06%
Apartment_v2	69.96%
Landmark_v2	81.35%

- **Observations:**

- Street names and building numbers were successfully extracted for the majority of addresses.
- Area details and landmarks had higher missing rates due to inconsistent input formats.
- This structured dataset provides a solid foundation for **estimating delivery locations** and **optimizing courier distribution** in subsequent steps.

2. Estimating Delivery Locations

Objective:

Since latitude and longitude values are missing for deliveries on **16-10-2025**, the goal is to estimate these geolocations using historical delivery data from **11-10 to 15-10-2025**. Each estimated location is accompanied by a **confidence score (0–100)**, indicating the reliability of the estimation.

Methodology:

1. **Data Preparation:**
    a. Historical deliveries (`df_hist`) were separated from the target date (`df_16`).
    b. Only the relevant columns, including `StreetFinal`, `BuildingNumber_v2`, and other structured address fields, were retained.
2. **Exact Street Match:**
    a. For deliveries whose street names exactly match historical streets, the mean latitude and longitude of that street from previous days were assigned.
    b. Confidence for exact matches was set to **95**.
3. **Similarity-Based Matching (TF-IDF + Cosine Similarity):**
    a. For unmatched addresses, **TF-IDF vectorization** of cleaned addresses was applied.

b. Cosine similarity was computed between 16-10 addresses and historical addresses.
c. The historical address with the **highest similarity** was selected as the match.
d. The confidence score was set proportional to the similarity (0–100).

4. **Fallback for Low Confidence (<50):**
    a. For addresses with confidence <50, the centroid (mean latitude and longitude) of the corresponding street from historical data was used.
    b. The minimum confidence after fallback was set to **50**.

Results:

Statistic	Value
Count	313
Mean	79.78
Std	20.03
Min	31
25%	62
Median	95
75%	95
Max	100

- Most deliveries have **high confidence estimates**, especially those with exact street matches.
- Low-confidence deliveries were effectively handled using fallback centroids.

Visualization:

- Histograms of confidence scores show a majority of deliveries with confidence ≥60, with the median at 95.

- After fallback, all deliveries have at least **50% confidence**, ensuring

Remarks:

- Combining **exact street matches** with **similarity-based matching** ensures robust estimation for missing geolocations.
- Confidence scores help identify deliveries that may require manual verification or additional data sources.
- These predictions will be used for **courier assignment optimization** in the next step.

3. Optimize Shipment Distribution to Couriers

Objective:
To efficiently assign deliveries to 20 couriers for **15-10-2025**, minimizing operational area while balancing the number of deliveries per courier.

Methodology:

1. **Data Preparation:**
    a. Extracted latitude and longitude of all deliveries.
    b. Confirmed **308 deliveries** for 15-10-2025.
2. **Initial Clustering with KMeans:**
    a. Applied **KMeans (n_clusters=20)** to group deliveries into 20 clusters.
    b. Each cluster represents an initial courier assignment.
3. **Operational Area Calculation:**
    a. Approximated cluster area using **Convex Hull** and converting latitude/longitude differences to km^2.
    b. **Total operational area (initial): 3.08 km^2**.
4. **Balancing Courier Load:**
    a. Couriers with **<10 or >20 deliveries** were identified.

b. Reassigned deliveries iteratively to nearby couriers, prioritizing:
    i. Minimizing distance from cluster centroids.
    ii. Keeping deliveries per courier within 10–20 range.
c. Repeated until all constraints satisfied.

5. **Results after Optimization:**

Courier	Deliveries	Area (km^2)
0	20	0.2129
1	17	0.1660
2	20	0.1661
...	...	...
19	20	0.0465

- **Total Operational Area:** 3.19 km^2
- **Min deliveries per courier:** 10
- **Max deliveries per courier: 20**

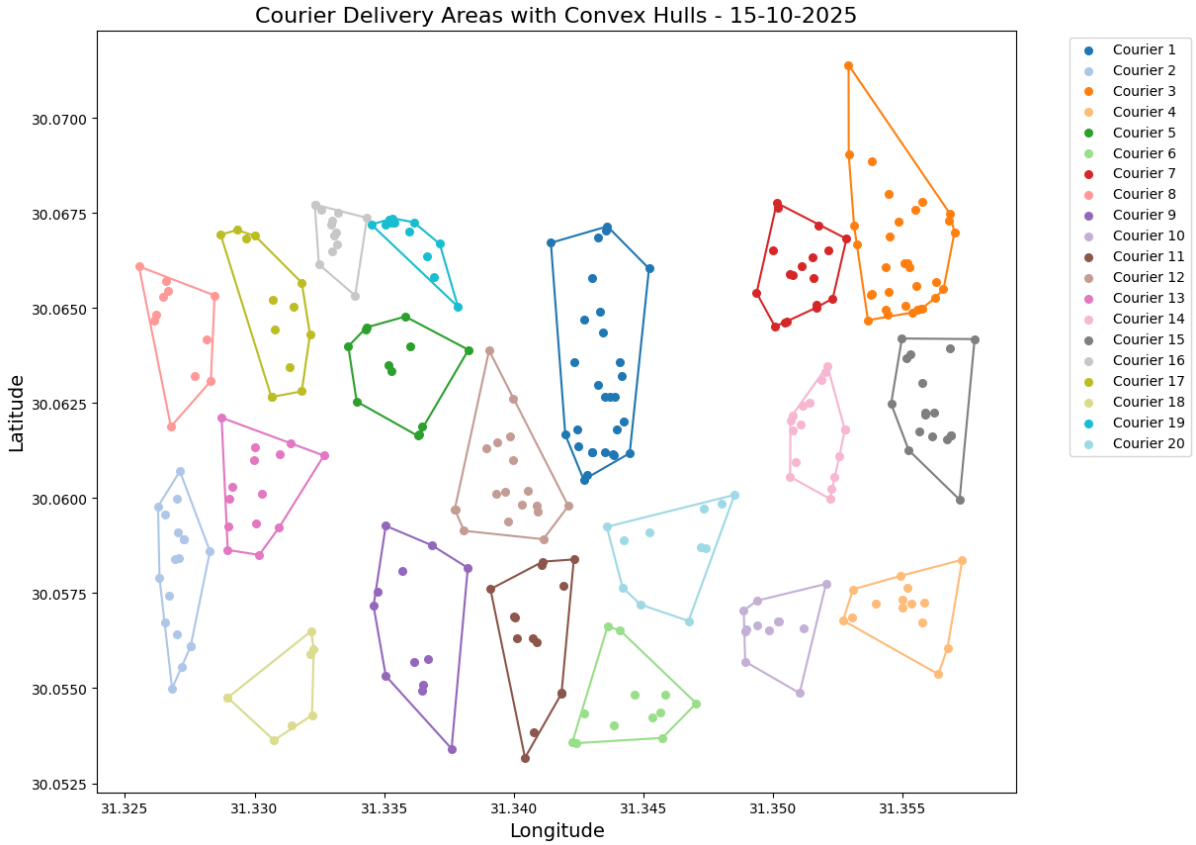
Alternative Optimization Attempt (Not Used in Final Assignment)

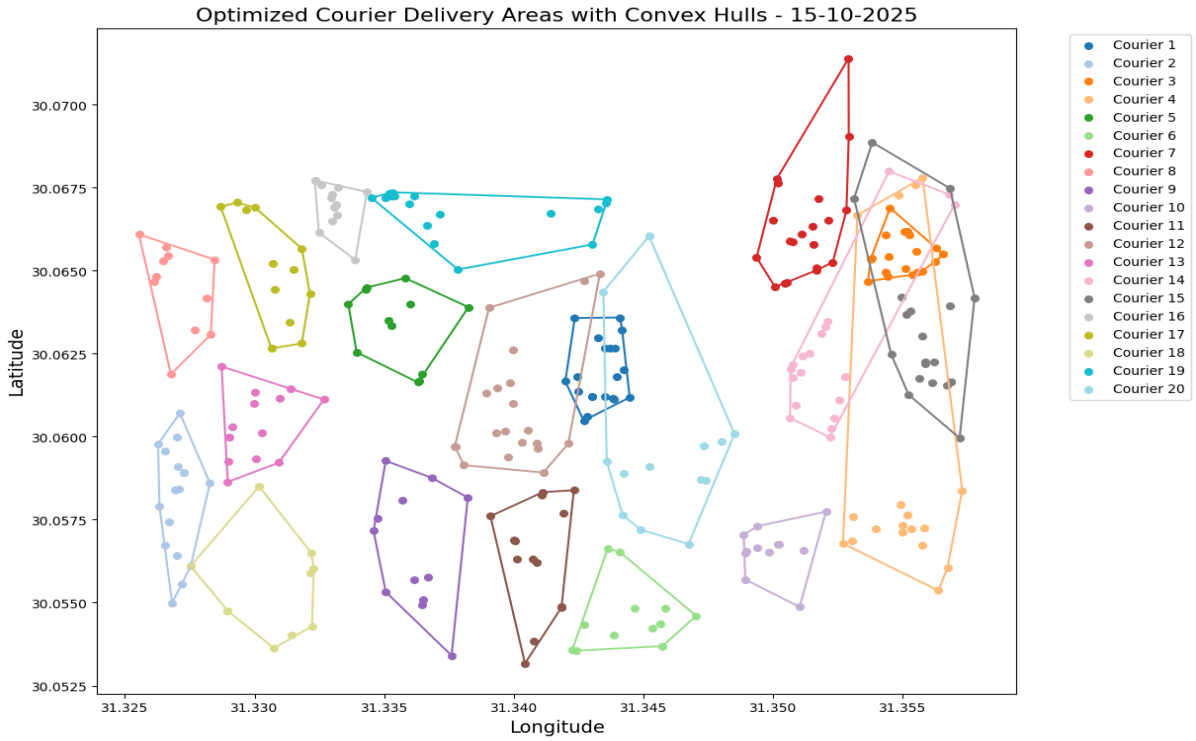
- A second approach was attempted to optimize delivery assignment to couriers:
    - **KMeans Clustering:**
        - Applied KMeans with 20 clusters based on delivery latitude and longitude.
    - **Iterative Adjustment:**
        - For couriers with more than 20 deliveries:
            - The farthest delivery from the cluster centroid was moved to the nearest underfilled courier (<20 deliveries).
        - For couriers with fewer than 10 deliveries:
            - Intended to receive nearby deliveries from overfilled couriers.
        - Iterations repeated multiple times to attempt balancing deliveries.
    - **Operational Area Calculation:**
        - Convex hulls were used to approximate the operational area per courier.
        - **Total Operational Area (km^2): 3.91 km^2** — notably higher than the final method (3.19 km^2).
- **Observation:**

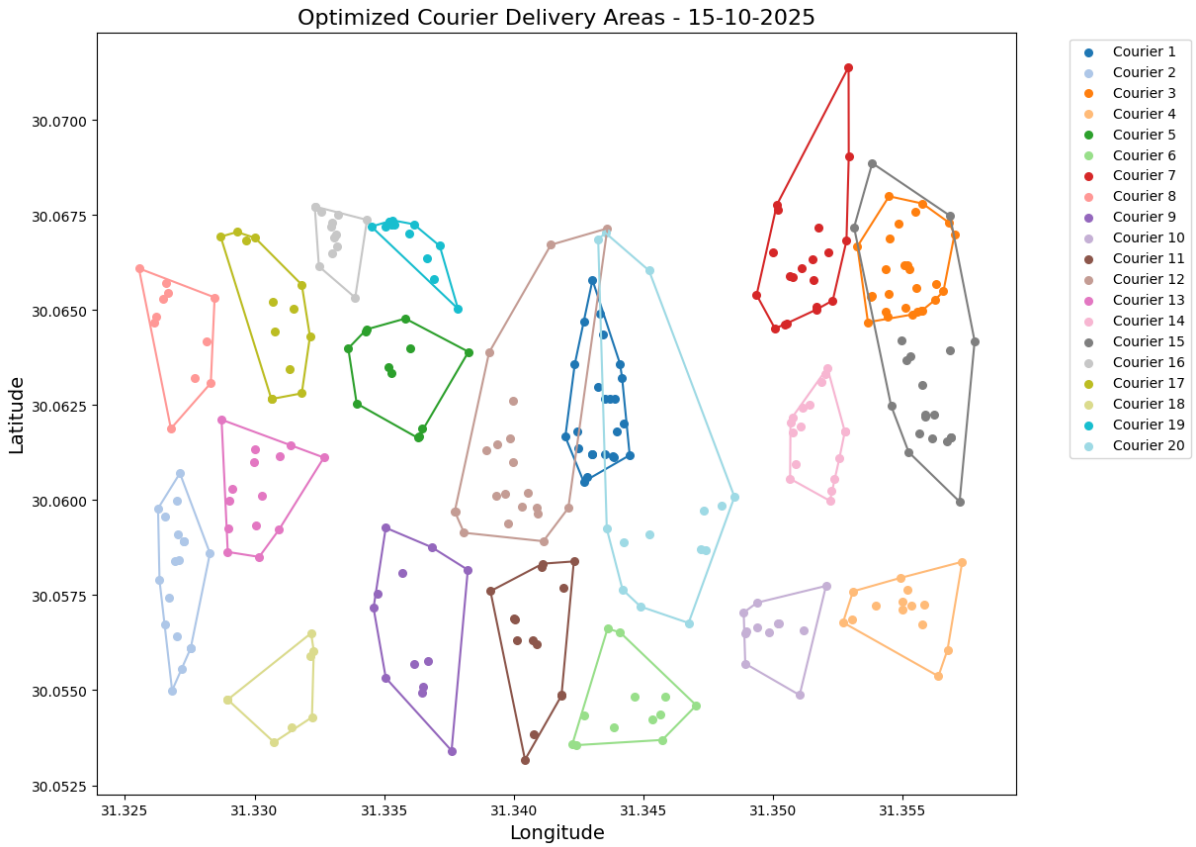
- Although delivery counts per courier were more balanced, the total operational area increased due to deliveries being reassigned to farther couriers.
- Therefore, this method was **not used for the final optimized assignment**.

Visualization:

- The following plot shows courier delivery areas after this attempt (Convex hulls in different colors).
- Note the larger operational spread compared to the final optimization.




Optimized Courier Delivery Areas with Convex Hulls - 15-10-2025
Latitude
Longitude
30.0700
30.0675
30.0650
30.0625
30.0600
30.0575
30.0550
30.0525
31.325
31.330
31.335
31.340
31.345
31.350
31.355
Courier 1
Courier 2
Courier 3
Courier 4
Courier 5
Courier 6
Courier 7
Courier 8
Courier 9
Courier 10
Courier 11
Courier 12
Courier 13
Courier 14
Courier 15
Courier 16
Courier 17
Courier 18
Courier 19
Courier 20


Optimized Courier Delivery Areas - 15-10-2025
Latitude
Longitude
30.0700
30.0675
30.0650
30.0625
30.0600
30.0575
30.0550
30.0525
31.325
31.330
31.335
31.340
31.345
31.350
31.355
Courier 1
Courier 2
Courier 3
Courier 4
Courier 5
Courier 6
Courier 7
Courier 8
Courier 9
Courier 10
Courier 11
Courier 12
Courier 13
Courier 14
Courier 15
Courier 16
Courier 17
Courier 18
Courier 19
Courier 20