EV CHARGING — New Site Selection: Step-by-step reference (notebook functions)

Prepared for: Your EV_Charging_Station_place_factory.ipynb workflow Role: Senior Data Scientist / AI Engineer — practical, copy-paste-ready instructions

PREREQUISITES - Notebook open: EV_Charging_Station_place_factory.ipynb - All required files and helpers loaded, including utils.py and geodataframes: * districts_gdf * district_coverage * poi_per_district (POI counts / layers) * gdf_full_updated (existing stations / sites geodataframe) * land_price, chargers_market_price, miscellaneous_costs (for investment calculation) - Ensure CRS for lat/lon is consistent (WGS84 / EPSG:4326) and distance functions expect km.

WORKFLOW (step-by-step — functions & purpose)

1) Place context: locate the candidate site (lat, lon) Function: place_2 = attach_site_context(lat=<LAT>, lon=<LON>, districts_gdf=districts_gdf, district_coverage=district_coverage, land_price=land_price) Purpose: Create a 1-row dataframe/geodataframe for the new site containing geometry, district, and other contextual columns.

2) Extract district Code: district_for_place_2 = place_2['district_name'].iloc[0] Purpose: Use district name for district-based cost assumptions and investment heuristics (e.g. land price tiers).

3) Compute distances from the new site to all existing sites Code: place_2_distances = calculate_new_site_distances(place_2, df_sites=gdf_full_updated) Purpose: Produce a dataframe/series of distances from the proposed site to each existing site (used by KNN predictions). Note: you mentioned ~1455 other sites — confirm len(gdf_full_updated) as sanity check.

4) Build the feature row for model / KNN prediction Code: place_2_df = build_df_place_to_predict_knn(place_2, num_of_chargers=2, charger_type='DC_fast') Purpose: Build a single-row dataframe (features expected by KNN prediction pipeline). Adjust num_of_chargers and charger_type for scenarios.

5) Add POI count (example within 2 km) Code: poi_counts_place_2 = poi_counts_within_2km(place_2_df, poi_per_district, radius_km=2.0) place_2_df['poi_within_2km'] = poi_counts_place_2['poi_within_2km'] Purpose: Capture local demand proxies (POI density) as a predictor.

6) Predict 6-month total volume — early / conservative (pre-ramp) Code: predicted_volume_place_2, neighbor_info_place_2, features_place_2 = predict_new_site_total_volume_knn(place_df=place_2_df, gdf_full_updated=gdf_full_updated, distance_long=place_2_distances, place_x=place_2, effective_radius_km=3, k_neighbors=40, power_weight=0.7, charger_scaling_exp=0.7) Interpretation: - Output (example): 495827.89255953726 - This is a **conservative early estimate** (ramp-up / pre-discovery) — useful for short-term cashflow planning.

7) Predict 6-month total volume — mature / weighted (post-ramp) Code: pred_volume_place_2_weighted, neighbor_info, features_used = predict_new_site_total_volume_knn_weighted(place_df=place_2_df, gdf_full_updated=gdf_full_updated, distance_long=place_2_distances, place_x=place_2, effective_radius_km=3, k_neighbors=40, charger_scaling_exp=0.6, power_weight=0.7, weight_same_type=0.7) Interpretation: - Output (example): 797187.8298368782 - This represents **steady-state / mature performance** — useful for long-run ROI and valuation.

8) Estimate investment Code: inv_place_2 = estimate_site_investment(district=district_for_place_2, area_sqm=100, charger_type='DC_fast', num_chargers=2, land_price=land_price, chargers_market_price=chargers_market_price, misc_costs=miscellaneous_costs,) total_investment_place_2 = inv_place_2['total_investment'] Example output: total_investment_place_2 = 3362934.75 Purpose: Financial planning — CAPEX / site set-up.

9) Calculate months to ROI Code (conservative / early): months_to_ROI_place_2_conservative = calculate_ROI(place_2, place_2_df, gdf_full_updated, total_investment_place_2, predicted_volume_place_2)

Code (after ramp-up / mature): months_to_ROI_place_2_after_rampup_period = calculate_ROI(place_2, place_2_df, gdf_full_updated, total_investment_place_2, pred_volume_place_2_weighted)

Example outputs: months_to_ROI_place_2_conservative = 34 months_to_ROI_place_2_after_rampup_period = 21

Purpose: Assess payback timing under early and mature demand assumptions.

SUMMARY TABLE (example values from your run) | Metric | Example value | Notes | |---|---|---| | Early predicted 6-month volume | 495,827.89 | conservative / ramp-up | | Mature

predicted 6-month volume | 797,187.83 | post-ramp, weighted KNN | | Total investment (CAPEX) | 3,362,934.75 | estimate from estimate_site_investment | | Months to ROI (conservative) | 34 | using early prediction | | Months to ROI (mature) | 21 | using post-ramp prediction | | Number of existing sites compared | ~1455 | length of gdf_full_updated (verify) | | Effective radius used (km) | 3 | neighbors selected within 3 km | | k_neighbors | 40 | KNN neighbor count | | num_chargers | 2 | candidate site equipment | | charger_type | DC_fast | affects power/prediction & cost |

PRACTICAL CHECKPOINTS & TIPS - Sanity checks: * Verify len(gdf_full_updated) == expected number of sites (e.g., 1455). * Visualize place_2 and nearest neighbors on a map to confirm geometry / distances. * Check POI density — if zero or extremely low, review poi_per_district prep steps. - Sensitivity analyses to run: * Vary num_chargers (1,2,4) and charger_type (AC_fast vs DC_fast). * Vary k_neighbors (20,40,60) and effective_radius_km (2,3,5) to measure model stability. * Vary charger_scaling_exp (0.5,0.6,0.7,0.8) and power_weight (0.5–0.9). - Business adjustments: * Early estimates should be used for short-term cash flow; mature estimates for valuation and investment decision-making. * Consider ramp-up curve: if you have historical ramp models, scale the mature estimate by estimated adoption percentage for the first 6 months. - Logging & reproducibility: * Save place_2_df, place_2_distances, neighbor_info and features_used for auditability. * Save the random seed (if functions use randomness) and function parameters in an experiment log.

COPY-PASTE TEMPLATE (example) ```python # 1. place context place_2 = attach_site_context(lat=40.12345, lon=-3.54321, districts_gdf=districts_gdf, district_coverage=district_coverage, land_price=land_price)

2. district district_for_place_2 = place_2['district_name'].iloc[0]

3. distances place_2_distances = calculate_new_site_distances(place_2, df_sites=gdf_full_updated)

4. build features row place_2_df = build_df_place_to_predict_knn(place_2, num_of_chargers=2, charger_type='DC_fast')

5. poi counts poi_counts_place_2 = poi_counts_within_2km(place_2_df, poi_per_district, radius_km=2.0) place_2_df['poi_within_2km'] = poi_counts_place_2['poi_within_2km']

6. predict (early) predicted_volume_place_2, neighbor_info_place_2, features_place_2 = predict_new_site_total_volume_knn(place_df=place_2_df, gdf_full_updated=gdf_full_updated, distance_long=place_2_distances, place_x=place_2, effective_radius_km=3, k_neighbors=40,

power_weight=0.7, charger_scaling_exp=0.7)

7. predict (mature / weighted) pred_volume_place_2_weighted, neighbor_info, features_used = predict_new_site_total_volume_knn_weighted(place_df=place_2_df, gdf_full_updated=gdf_full_updated, distance_long=place_2_distances, place_x=place_2, effective_radius_km=3, k_neighbors=40, charger_scaling_exp=0.6, power_weight=0.7, weight_same_type=0.7)

8. investment inv_place_2 = estimate_site_investment(district=district_for_place_2, area_sqm=100, charger_type='DC_fast', num_chargers=2, land_price=land_price, chargers_market_price=chargers_market_price, misc_costs=miscellaneous_costs,) total_investment_place_2 = inv_place_2['total_investment']

9. ROI months_to_ROI_place_2_conservative = calculate_ROI(place_2, place_2_df, gdf_full_updated, total_investment_place_2, predicted_volume_place_2) months_to_ROI_place_2_after_rampup_period = calculate_ROI(place_2, place_2_df, gdf_full_updated, total_investment_place_2, pred_volume_place_2_weighted) ```

END OF DOCUMENT

Summary table — example values

Metric	Value	Notes
Early predicted 6-month vo	495,827.89	conservative / ramp-up
Mature predicted 6-month v	797,187.83	post-ramp, weighted KNN
Total investment (CAPEX)	3,362,934.75	estimate from estimate_site_investmen
Months to ROI (conservative	34	using early prediction
Months to ROI (mature)	21	using post-ramp prediction
# existing sites compared	~1455	len(gdf_full_updated) (verify)
Effective radius (km)	3	neighbors selected within 3 km
k_neighbors	40	KNN neighbor count
num_chargers	2	candidate site equipment
charger_type	DC_fast	affects power/prediction & cost