Task Report: Warehouse.

Data preparation and cleaning

In the cleaning stage I first fixed the data types by turning number-like text columns into real numbers, parsing all timestamps into CET datetimes, and giving the columns clear names.

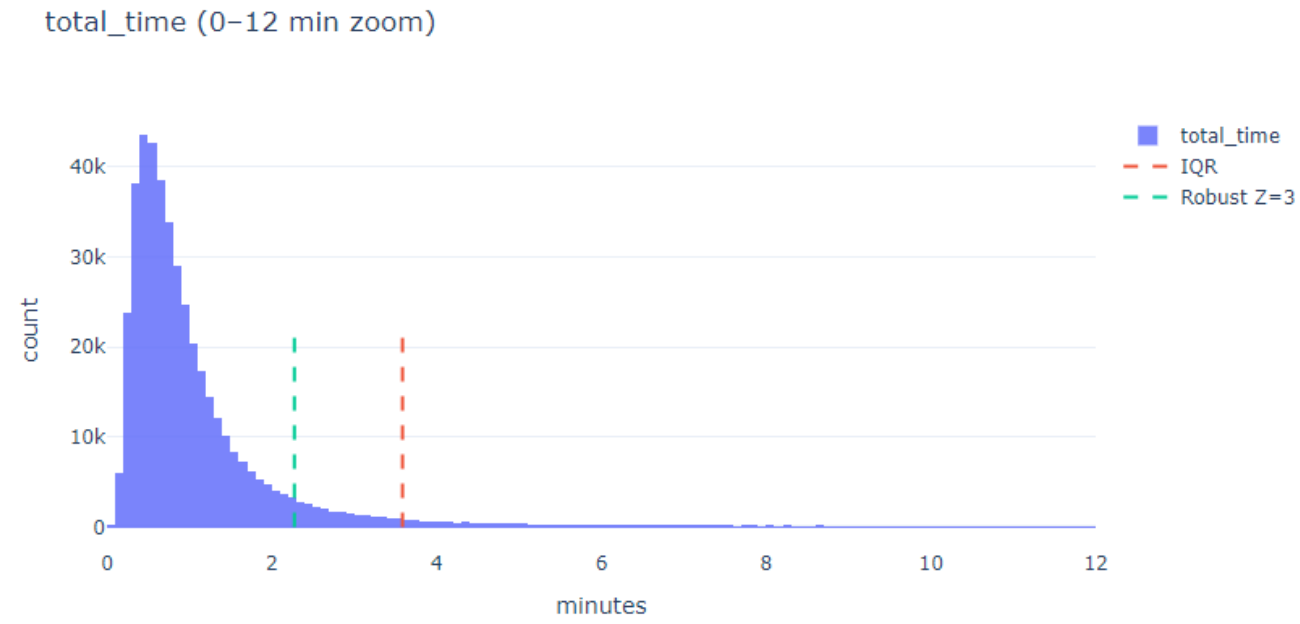
I then created some helper variables: **total_time** as the sum of pick, **mezipick** and **nopick** minutes, the shares of these three parts, and the actual handling time between start and end.

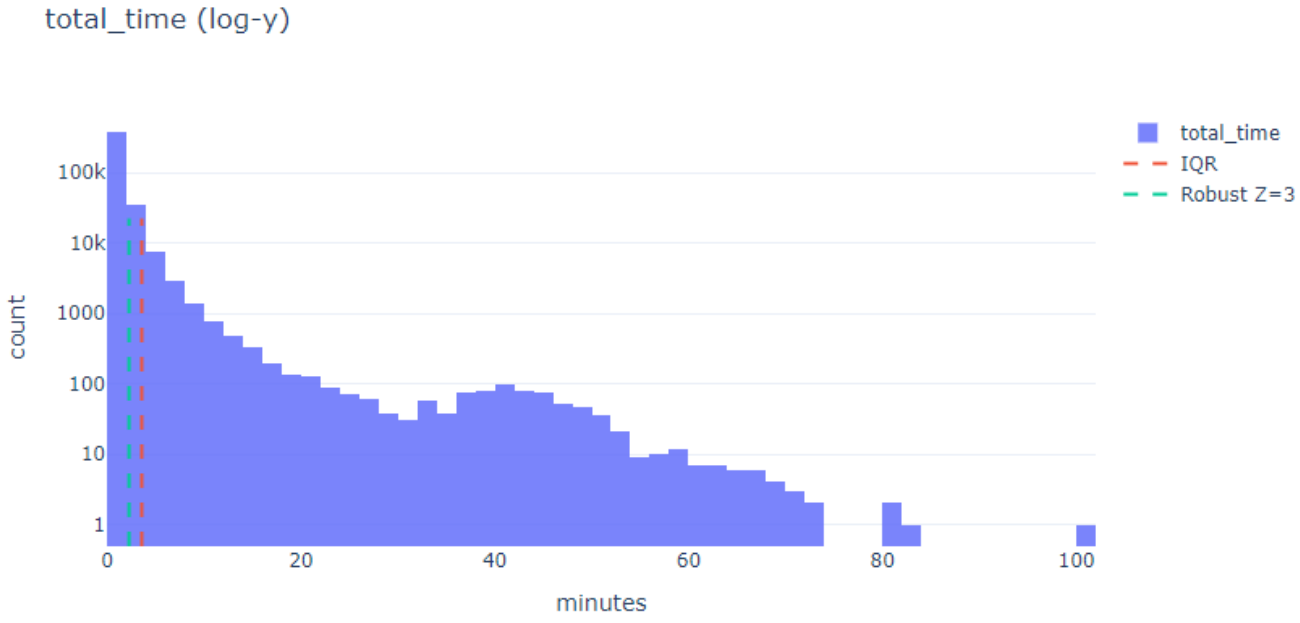
I checked missing values for every column and saw that some fields, like minut_**mezipick** and **minut_nopick**, were often empty, which makes sense because not every operation needs them. To find suspicious rows I flagged cases where a positive quantity had zero time, where the end time was before the start, or where the duration was negative; these should be removed, however there were none.

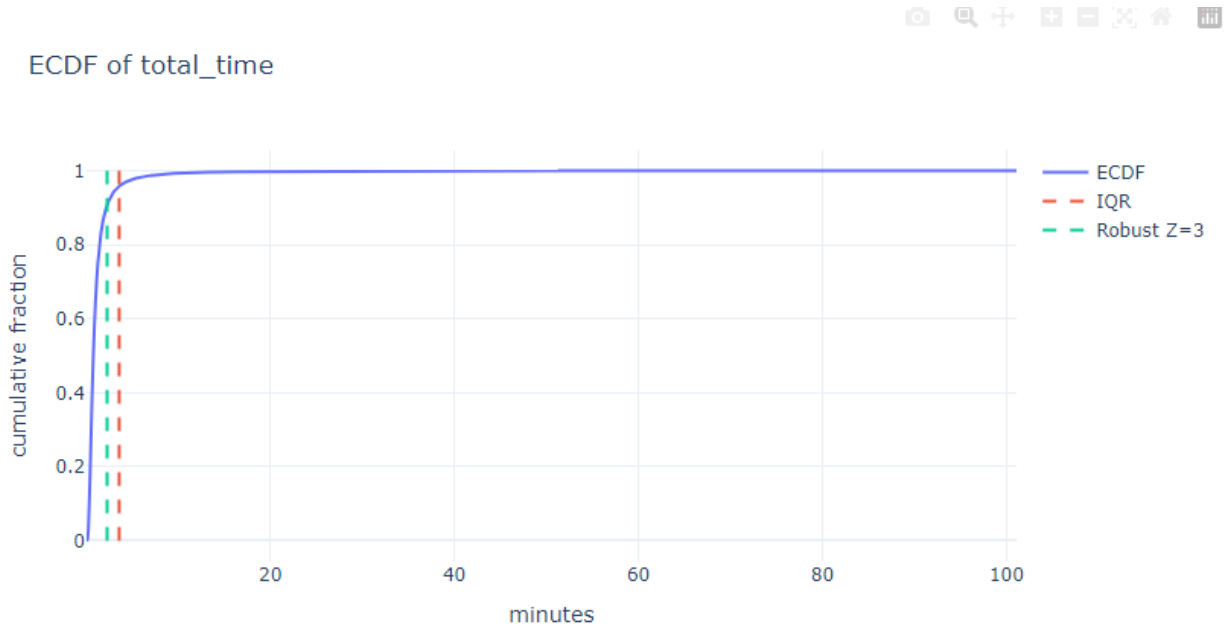
For outliers in **total_time**, which has a long skewed tail, I compared two methods: one based on the interquartile range (Q3 + 3*IQR) and another using a robust Z-score from the median and median absolute deviation.

I plotted histograms, log views and ECDF curves to see how the thresholds cut the data. Finally, I chose the robust Z-score method as the main rule, which keeps most realistic operations but removes clear errors and extreme cases.

I sanity checked my assumptions on the names of the columns with ChatGPT


total_time (0–12 min zoom)
total_time
IQR
Robust Z=3
40k
30k
20k
10k
0
count
0
2
4
6
8
10
12
minutes


total_time (log-y)
total_time
IQR
Robust Z=3
100k
10k
1000
100
10
1
count
0
20
40
60
80
100
minutes



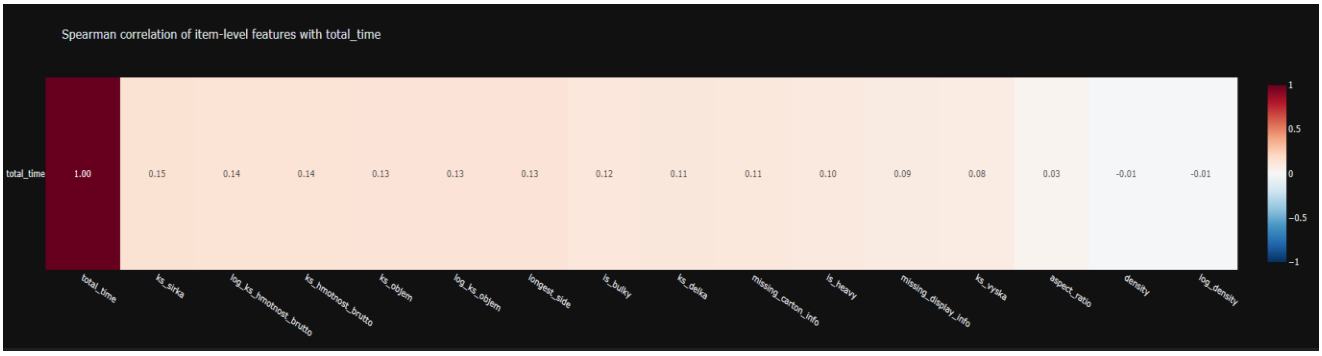
Feature selection and engineering

In the feature selection and engineering step I tried to understand which aspects of the data are most strongly related to operation speed and to create new, more informative variables from the raw inputs.

I started with simple **time-based features** such as hour of day, day of week, and weekend or peak-hour flags, and checked their correlation with picking time; while individual correlations were small, they provide context about working conditions.

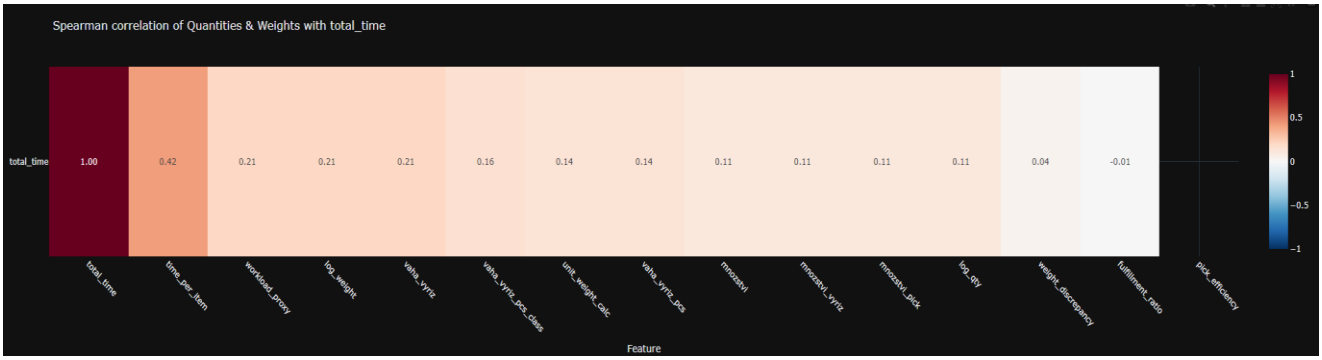
Location features such as workstation, warehouse zone, and processing method were handled as categorical variables, capped to the most frequent values to avoid over-fragmentation, and evaluated with mutual information; this showed that certain workstations and zones systematically influence operation duration.

Item-level physical attributes were transformed into more descriptive measures like density, longest side, aspect ratio, and binary indicators for “bulky” or “heavy,” and correlations confirmed that larger, heavier items generally take longer to process.



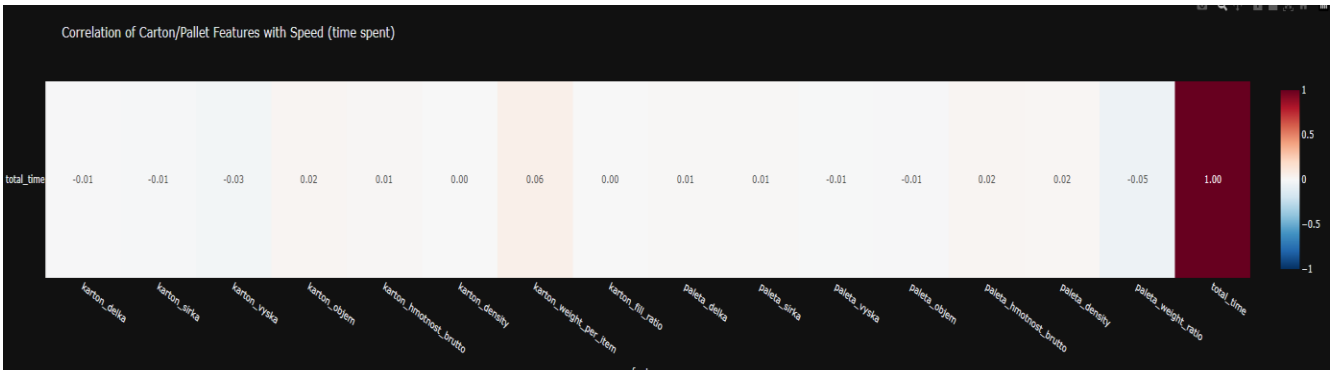
At the **workload** level I engineered variables such as fulfillment ratios, pick efficiency, per-unit weight, workload proxies, log-transformed weights and quantities, and normalized time per item, which together capture how task size and efficiency affect speed.

These showed stronger correlations with total time than raw quantities alone. Specifically, logically the **time_per_item, workload_proxy, log_weight and vaha_vyriz**



For **product hierarchy**, direct use of high-cardinality category codes was not practical, so I aggregated median speeds and counts per hierarchy node and highlighted groups with systematically higher times, visualized as a sunburst tree of slow product families.

Finally, I extended the analysis to packaging and logistics, deriving carton and pallet densities, weight ratios, and fill ratios, which on paper should explain inefficiencies linked to packaging formats. However the results were not significant.

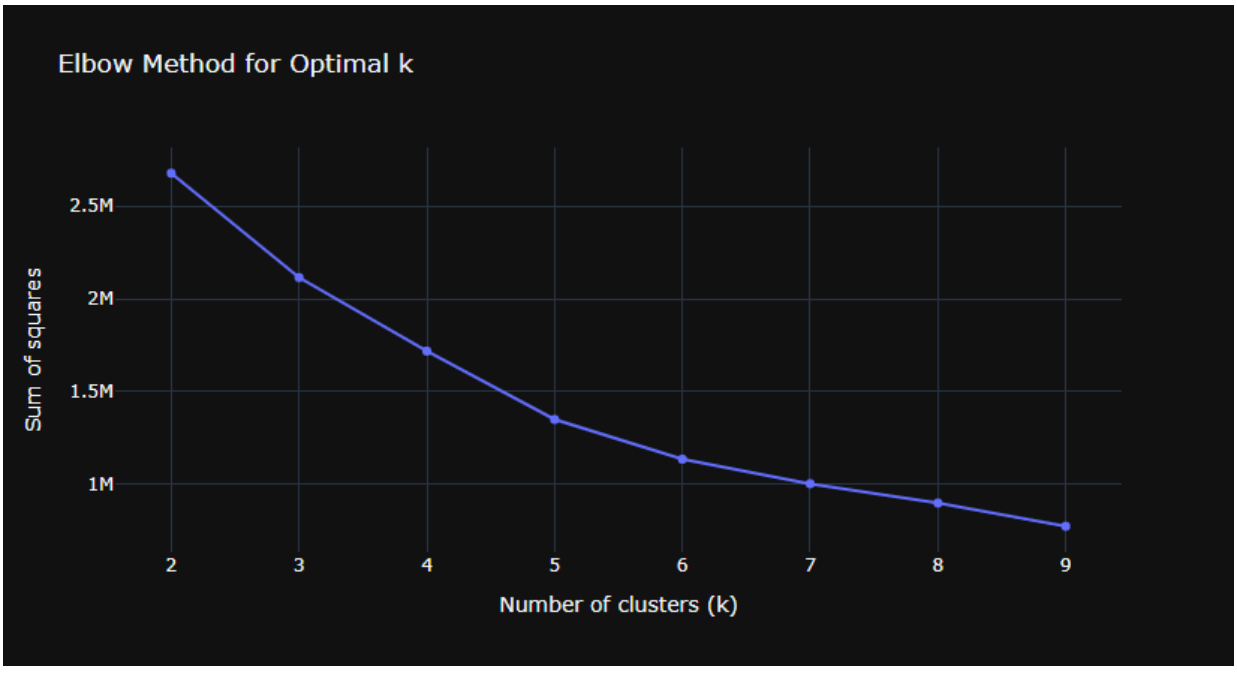


All these engineered features were saved in structured parquet datasets by theme (time, workload, physical, location, hierarchy, packaging), giving a clear modular base for clustering and further modeling.

Clustering & Interpretation

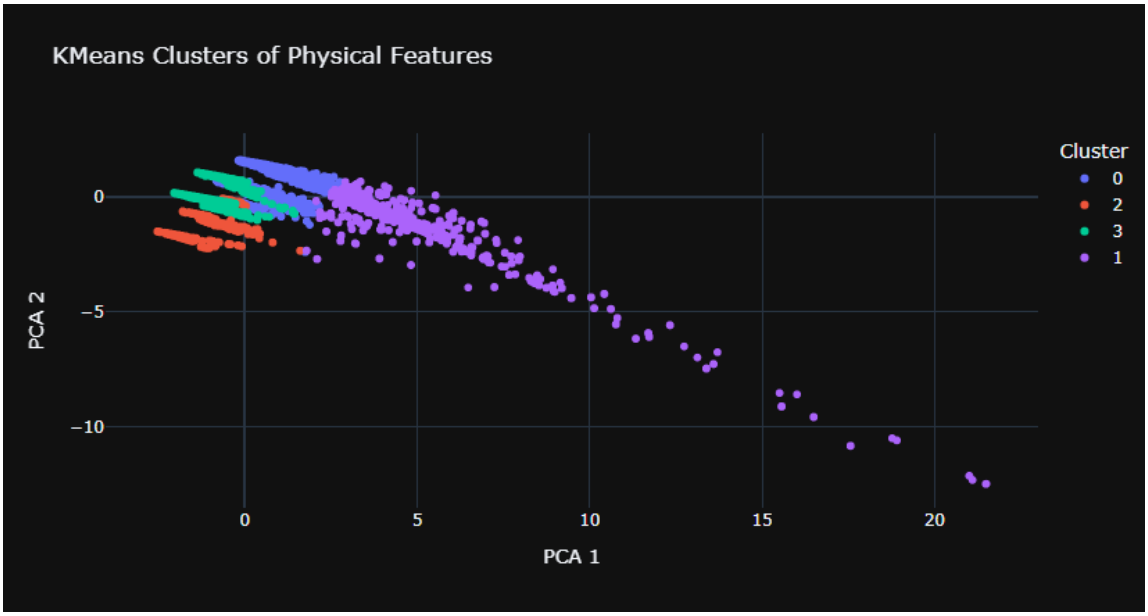
In the clustering stage the goal was to group operations into categories that are both internally consistent and large enough to be meaningful. I applied **KMeans clustering** across several feature sets, always starting by scaling numerical features with StandardScaler (so that size, weight, and workload are comparable on the same scale) or one-hot encoding categorical features (so workstation and zone codes could be treated numerically).

To decide the number of clusters, I used the **elbow method** on the inertia curve and tested values of k in the range of 2–10; for most domains k≈3–4 provided the best trade-off between compact groups and interpretability.



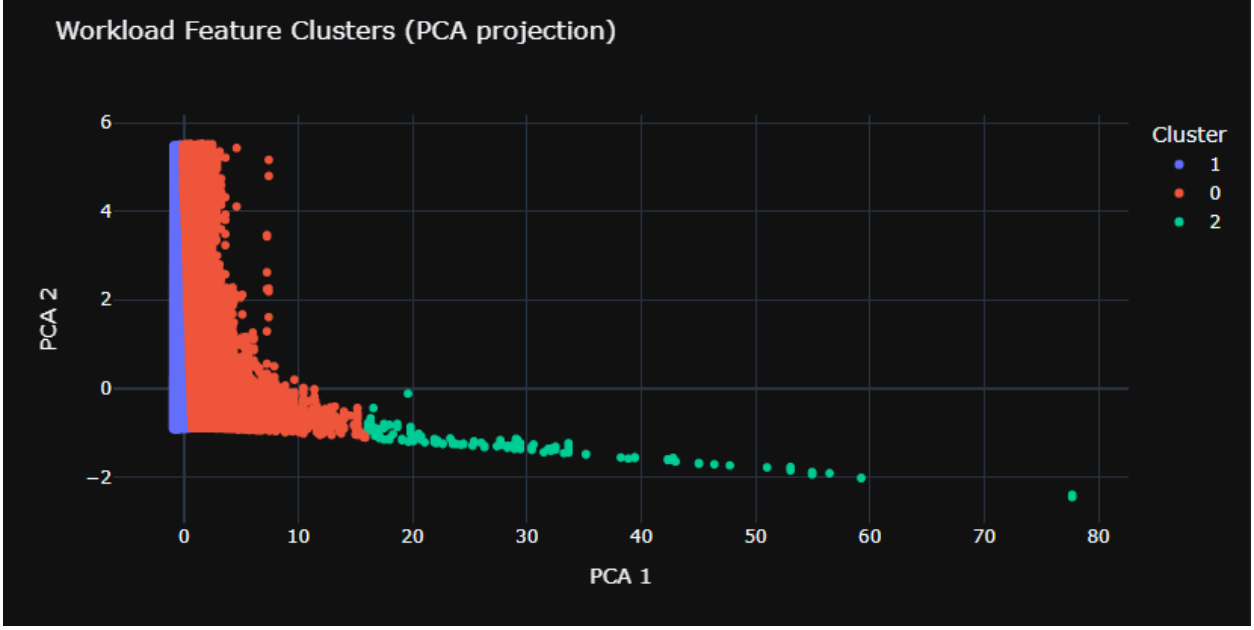
To check quality, I compared median **total_time** across clusters and visualized the distributions with boxplots, and also projected high-dimensional clusters into two dimensions with **PCA** for inspection.

The results were consistent across domains. When clustering on **physical attributes** (size, weight, volume, “bulky” flags), clusters separated into small/light items with fast processing, medium items with average times, and bulky/heavy items with systematically longer handling times.

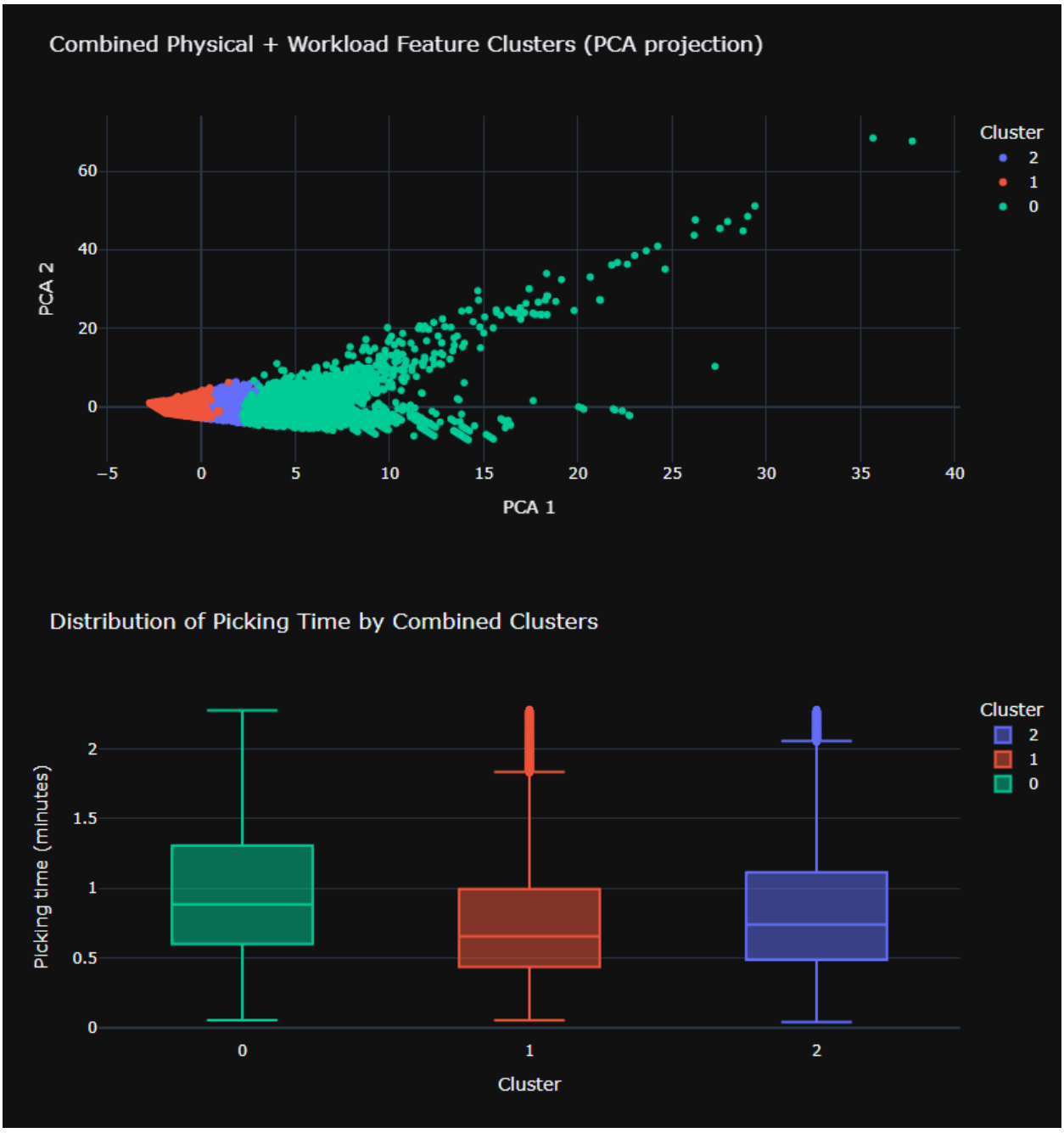


Using **workload features** (quantities, weights, time per item), clusters split tasks into high-volume orders (longer but efficient on a per-item basis), medium-scale orders, and low-volume but irregular tasks where time per item was higher.


Distribution of Picking Time by Workload Cluster
Picking time (minutes)
2
1.5
1
0.5
0
0
1
2
Cluster
Cluster
1
0
2


Workload Feature Clusters (PCA projection)
PCA 2
6
4
2
0
−2
0
10
20
30
40
50
60
70
80
PCA 1
Cluster
1
0
2

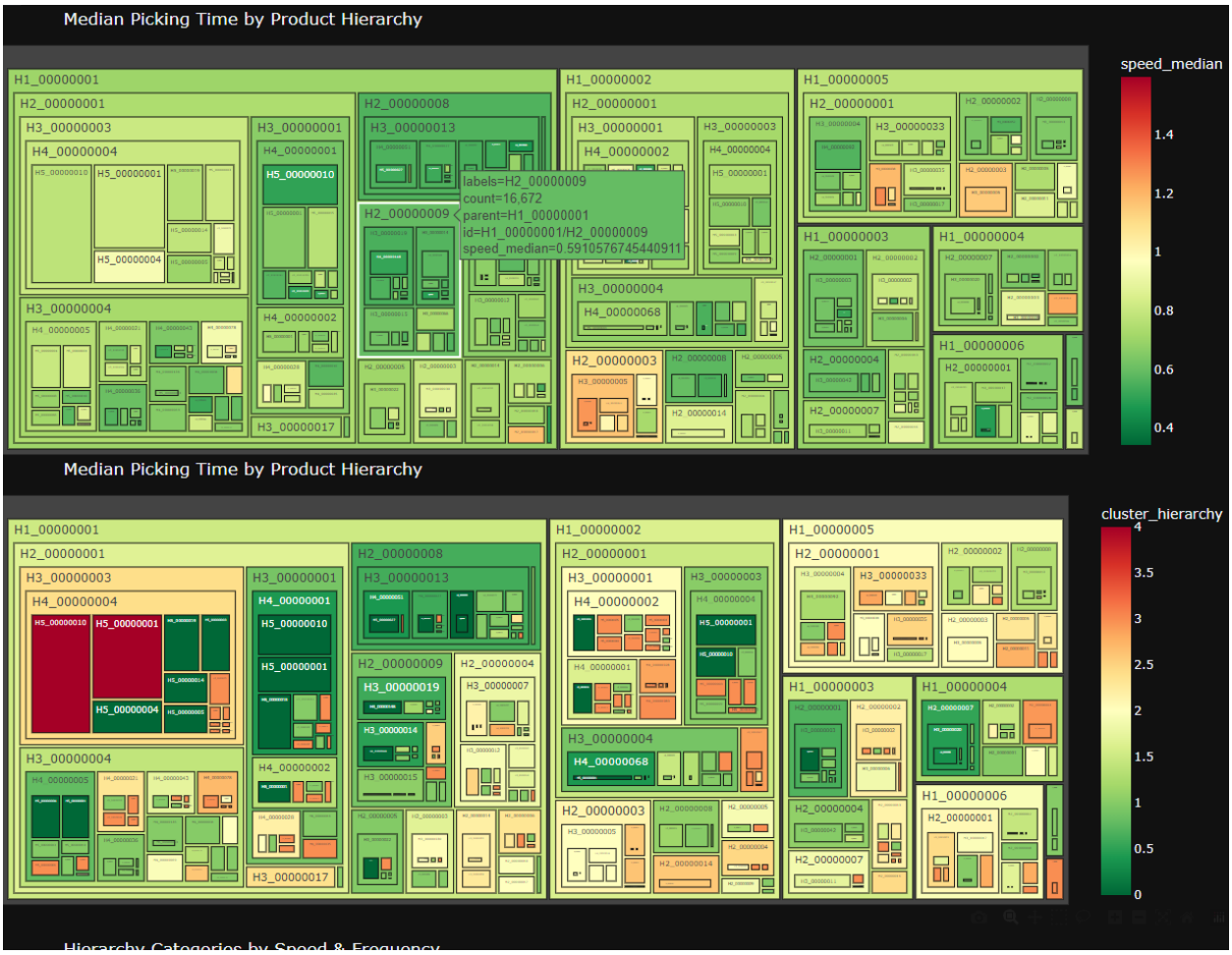
Combining **physical and workload features** produced a more robust segmentation that captured both item difficulty and order size, showing clusters of light/small orders (fast), standard medium orders, and heavy/bulky orders with long durations.



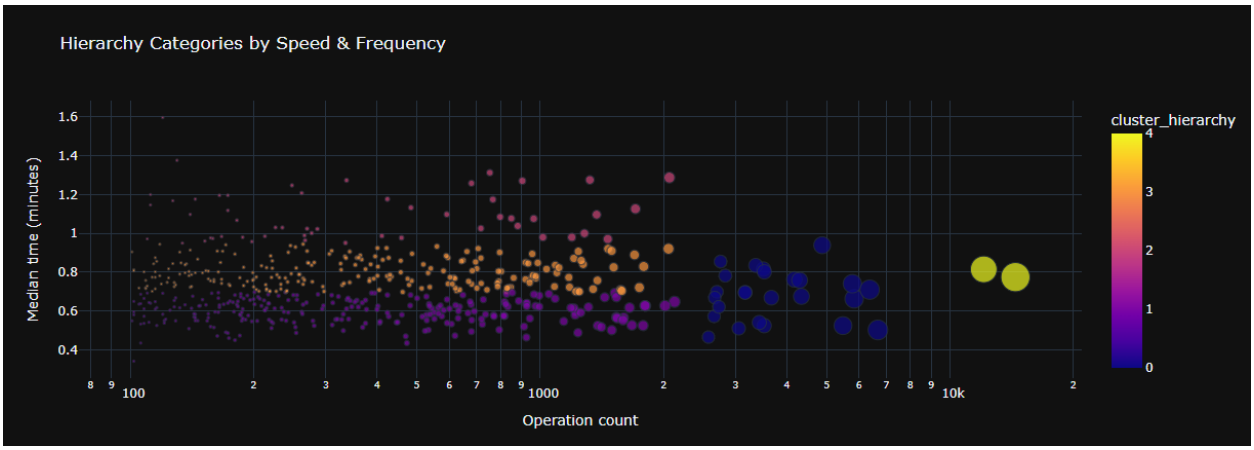
When judging the weight of the items, we can see that the heavy items have higher deviances, they are less homogeneous due to them being more rare and special cases.

Clustering by **location and workstation codes** revealed that some zones and processing routes are associated with much slower operations, confirming that layout and workflow strongly affect speed. What it practically means, is that we cannot neglect the effect and organization of the workspace on the performance of the branch. The workers are dealing with differently structured warehouses, therefore different workload.

Finally, clustering at the level of the **product hierarchy** required aggregating to median speeds and counts per hierarchy path; here, clusters distinguished fast-moving families (frequent, low time), standard families, and rare product groups that are consistently slow. Treemaps colored by cluster helped to navigate this hierarchy, while scatter plots of speed versus frequency explained the separation.



The second figure might seem a little less transparent, but it just outlines that even under different categories we have products that behave similarly during a task execution.



On the scatter plot we can clearly see the usual clustering vizualization.

Proposals

The analysis shows that not all warehouse operations are equal: some are fast and routine, while others are naturally slower because of item size, workload scale, or location in the warehouse. Based on the clustering results, several improvements can be suggested:

1. **Handle bulky and heavy items separately.** These items consistently form the slowest clusters. They should be picked in dedicated zones or with proper equipment so they do not slow down regular orders.
2. **Reorganize warehouse layout.** Some areas and routes are clearly slower than others. Frequently picked items, especially those that are heavy, should be moved closer to dispatch areas. Routes that force long travel times should be optimized. Maybe is makes sense to standardize the warehouses design so we can take out its effect out of the equation.
3. **Use fair performance standards.** It is not realistic to expect the same picking speed for small, light items and for bulky, heavy ones. Standards should be set by cluster: fast items with low times, medium items with average times, and bulky items with longer times. This creates fairer KPIs and avoids penalizing workers for difficult tasks.
4. **Improve packaging consistency.** Variations in carton and pallet density slow down handling. Encouraging more standardized packaging would reduce inefficiency and make picking more predictable. Although the effect is negligible.
5. **Monitor and update regularly.** The clustering should be repeated over time to see if changes in layout, training, or packaging are improving results. Over time, we would expect clusters to become more balanced and less extreme.

Bonus points

Interpretation of the individual components

MINUT_PICK, MINUT_MEZIPICK, and MINUT_NOPICK are affected differently.

My assumption:

MINUT_PICK reflects direct handling effort and scales with item size and weight.
MINUT_MEZIPICK reflects travel between picks and is more influenced by warehouse layout and routing.

MINUT_NOPICK captures delays (waiting, searching, interruptions). It appears less systematic (has a lot of nans), and probably creating outliers.

New small clusters

When clustering, a few small groups naturally appeared with very few observations. These small clusters are often too rare to set formal standards. The recommended approach is either to merge them with their closest larger cluster (if they share similar characteristics) or to ignore them in KPI setting while still monitoring them separately. However, non-standardized processes might result in process bottlenecks as there are no clear guidelines on dealing with them.

Meaningful cluster characteristics

The clusters can be described in practical terms rather than abstract numbers:

1) *Light and small orders*: low times, efficient handling.
2) *Standard orders*: medium weight/volume, average times.
3) *Bulky/heavy orders*: systematically slower.
4) *Slow locations*: specific workstations or high shelves with higher times.
   These labels make the results easy to communicate and link back to warehouse operations.
5) *Similarly*-behaving product types: although the categorization is in-place, for the warehouse optimization it makes sense to revise the categories based on processing efficiency.

Maintaining the standards

Standards should not be revised on a periodic basis as they change with time. As warehouse layout, packaging, or staff change, new data should be fed into the same analysis pipeline.

Running this clustering quarterly or biannually allows management to update benchmarks, ensure fairness, and quickly detect if certain product groups or zones are drifting away from expected performance.