42577 Introduction to Business Analytics

Sergi Lupon Santacana
s254311@dtu.dk
s254311

Matteo D'Souza
s242947@dtu.dk
s242947

Marta Arana Pareja
s254310@dtu.dk
s254310

Esben Søvndahl Kok
s254322@dtu.dk
s254322

Report Contribution: All team members contributed equally to the organization of the project. **Matteo D'Souza** led the prediction modelling and clustering analysis; **Esben Søvndahl Kok** led the repositioning strategy and weather sensitivity analysis; **Marta Arana Pareja** contributed to the exploratory component; and **Sergi Lupon Santacana** contributed to data analysis and report preparation.

1. Introduction & Motivation

Bike-sharing systems offer cities an emission-free mobility option that is both efficient and attractive. But making these systems work well is harder than it looks. The core operational challenge is simple to state: bikes need to be where riders want them. If morning commuters drain stations in residential areas and flood stations in business districts, the system fails users in both directions.

This paper tackles that challenge using data from Citi Bike, one of the largest bike-sharing systems in the United States. We analyze over 17.5 million trips across 850 stations from 2018, with the goal of predicting hourly demand and translating those predictions into actionable repositioning strategies.

Our approach has three parts. First, we cluster stations spatially so that nearby stations are grouped together. Second, we build and compare multiple prediction models to forecast next-day hourly pickups and dropoffs for each cluster. We start with simple linear baselines and work up to ensemble methods like Random Forest and Gradient Boosting, paying attention to the bias-variance tradeoffs along the way. Third, we evaluate whether our predictions actually help with the overnight repositioning problem: can we use predicted demand to figure out how many bikes each cluster needs at the start of the day?

Beyond the prediction challenge, we investigate an exploratory question that emerged from our modelling work. Weather features turned out to have minimal impact on our predictions, which surprised us given conventional wisdom about outdoor activities. We dig into why: does weather sensitivity vary by cluster type? Commuters have to get to work regardless of rain, but recreational riders can stay home. If this hypothesis holds, it would explain our finding and clarify when weather-based operational adjustments are actually worthwhile.

The analysis reveals both the power and the limits of predictive approaches for bike-sharing operations, and highlights cases where simpler methods might be the better choice.

2. Data Analysis and Visualization

The 2018 NYC Citi Bike dataset contains 17,548,339 trips across 850 stations spanning the entire calendar year. Each trip record includes timestamps, station locations, user demographics, and trip duration. Our first task was understanding what this data could tell us about bike-sharing demand patterns.

Before proceeding with the analysis, we identified and removed rides that were located outside of NYC or had an unknown station, leaving 834 stations for analysis.

In addition, we discarded rides with very high trip durations in Fig. 1. These rides lasted more than $2 \times 2 \times 10^7$ (several months), likely data errors or lost/stolen bikes rather than actual trips. The boxplot shows a clear break between normal trips (under 1 hour) and extreme outliers. We removed the upper 5% of the distribution, cutting 876,004 entries while keeping legitimate trips intact.

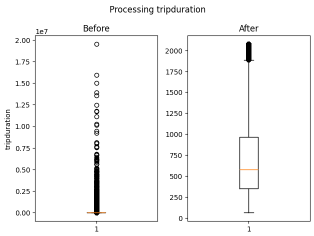

Figure 1. Distribution of trip durations before and after outlier removal.

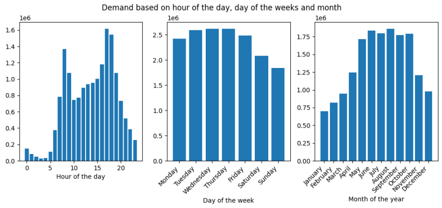

Figure 2. Demand patterns by hour, day of week, and month. Clear commuter patterns emerge with peaks at 8am and 5-6pm on weekdays.

After cleaning, the dataset contains 16,672,335 trips across 834 stations. The median trip duration is 7 minutes, with the distribution heavily right-skewed toward longer recreational rides.

To characterise bike-sharing demand patterns over time, we analysed the temporal structure of the data (Fig. 2). The temporal patterns reveal a story of urban commuting. Hourly demand shows pronounced morning (8am) and evening (5-6pm) peaks on weekdays, characteristic of work commutes. Weekend patterns flatten, with gradual builds from late morning through afternoon. Weekday ridership dominates, with weekends showing 30-40% lower demand. Monthly patterns reveal strong seasonality: ridership peaks in summer (June-September) and drops significantly in winter months.

To support the spatial clustering task in Section 3, we explored optimal cluster numbers using the elbow method. By computing the inertia (within-cluster sum of squared distances) for k ranging from 20 to 100, we identified the point of diminishing returns.

Figure 3 shows a steep decrease in inertia for small k, followed by a clear flattening around $k = 40$, indicating diminishing returns beyond this point. This analysis informed our choice of 40 clusters for the spatial clustering approach described in Section 3.

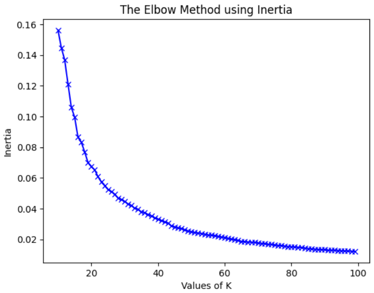

Figure 3. Application of elbow method to determine optimal K value for K-means clustering.

3. Prediction Challenge

3.1. Spatial Clustering

Our bike-sharing demand forecasting approach starts by clustering stations purely based on their geographic proximity: stations with latitude–longitude coordinates that are closer in space are grouped together. To build these clusters, two clustering algorithms were considered: K-means and DBSCAN. K-means partitions the stations into k non-overlapping clusters by iteratively assigning each station to the nearest centroid and then updating the centroids as the mean of the assigned points; this produces compact, roughly spherical clusters, which is appropriate when demand varies smoothly over space and clusters are expected to be of comparable size.

Based on the elbow method analysis in Section 2, which identified $k = 40$ as the point of diminishing returns, we chose this value for K-means clustering.

DBSCAN, in contrast, is a density-based method that groups stations wherever local point density exceeds a threshold and labels isolated stations as noise. With parameters *min_samples* = 3 and $\varepsilon = 0.0033$, DBSCAN can discover irregularly shaped clusters and identify anomalous stations, helping address K-means limitations (convex clusters, sensitivity to outliers and spatial heterogeneity). To compare the two algorithms, the silhouette score was used as the main metric; this score ranges from -1 to 1 and measures how similar a station is to its own cluster compared with other clusters: values close to 1 indicate compact, well-separated clusters, values around 0 suggest overlapping clusters, and negative values imply that many points may be misassigned. Using this metric, K-means with $k = 40$ achieved a silhouette score of approximately 0.35, whereas DBSCAN with *min_samples* = 3 and $\varepsilon = 0.0033$ attained a negative score of roughly -0.16; as summarised in Table 1, these values show that K-means yields substantially more coherent and interpretable clusters of nearby stations than DBSCAN in this setting.

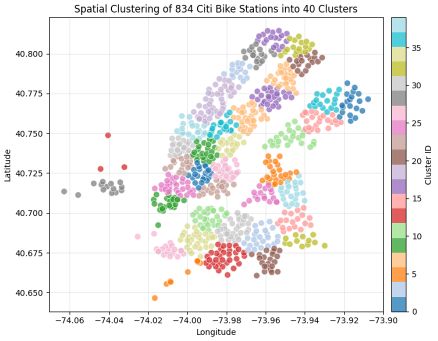

Figure 4. Spatial distribution of 40 K-means clusters.

Algorithm	Silhouette
K-Means (k = 40)	0.3516
DBSCAN (min_samp = 3, eps = 0.0033)	-0.1553

Table 1. Silhouette scores for clustering algorithms.

We therefore applied K-means with $k = 40$ to cluster the stations. For this analysis, we focused on three representative clusters selected to span the full range of demand patterns: cluster 9 (highest total demand), cluster 24 (median demand), and cluster 5 (third-lowest demand with sufficient data for analysis; the bottom two clusters had insufficient observations). This stratified selection allows us to assess model performance across the demand spectrum while keeping the analysis tractable.

3.2. Demand Prediction

To formulate a solution for predicting the next 24 hours of demand in terms of pickups and dropoffs, several experiments, models, and configurations were estimated for the previously selected clusters. Their predictive performance was assessed using three standard regression metrics on both training and test sets: R^2, RMSE, and MAE. The coefficient of determination R^2 measures the proportion of variance in the target variable that is explained by the model, with values closer to 1 indicating a better fit. RMSE (root mean squared error) and MAE (mean absolute error) both quantify the average magnitude of the residuals in the same units as the target; RMSE is computed from squared residuals and therefore penalises large errors more strongly than MAE, whereas MAE weights all errors linearly. Considering these metrics together allows a fair comparison between models that may balance fit and generalisation differently for each specification reported in Tables 2 (pickups) and 3 (dropoffs).

The modelling of our problem began with the linear baselines in Tables 2 and 3 (“Linear Regr.” and “Lagged Regr., lag=1d”). Plain Linear Regression underfit badly: it delivered low R^2 and high RMSE/MAE on both train and test splits, indicating a model with high bias that could not

Model	Configuration	Split	R^2	RMSE	MAE
Linear Regr.	–	train	0.089	108.75	74.05
		test	0.030	88.30	66.37
Lagged Regr.	lag=1d	train	0.772	54.46	29.75
		test	0.684	50.41	28.63
Random Forest	100 est.; feats=all	train	0.995	8.15	3.75
		test	0.671	51.44	22.27
Random Forest	200 est.; depth=30; leaf=17; split=2; feats=6; samp=0.5	train	0.929	30.35	14.55
		test	0.764	43.56	21.82
Lagged RF	lag=1d; 200 est.; depth=30; leaf=17; split=2; log2 feats; samp=0.5	train	0.892	37.57	18.31
		test	0.799	40.29	20.59
Gradient Boosting	lag=1d; 200 est.; 6 feats	train	0.894	37.19	19.67
		test	0.806	39.45	22.01

Table 2. Model performance for predicting **pickups**. Abbreviations (used in both tables): **est.** = number of estimators; **depth** = max tree depth; **leaf** = minimum samples per leaf; **split** = minimum samples to split a node; **feats** = max features; **log2 feats** = max features set to $\log_2(\text{n\_features})$; **samp** = subsample ratio; **lag=1d** = one-day lagged features.

Model	Configuration	Split	R^2	RMSE	MAE
Linear Regr.	–	train	0.067	109.09	76.06
		test	0.017	91.20	68.28
Lagged Regr.	lag=1d	train	0.765	54.81	29.80
		test	0.695	50.82	28.79
Random Forest	100 est.; feats=all	train	0.995	7.88	3.71
		test	0.705	49.96	21.70
Random Forest	200 est.; depth=30; leaf=17; split=2; feats=6; samp=0.5	train	0.924	31.07	15.20
		test	0.739	47.01	23.75
Lagged RF	lag=1d; 200 est.; depth=30; leaf=17; split=2; log2 feats; samp=0.5	train	0.891	37.27	18.30
		test	0.795	41.64	21.37
Gradient Boosting	lag=1d; 200 est.; 6 feats	train	0.896	36.45	19.01
		test	0.827	38.25	21.08

Table 3. Model performance for predicting **dropoffs**. Abbreviations as in Table 2.

capture the non-linear, temporal structure of cluster-level demand.To reduce bias in the linear regression model for pickups and drop-offs, temporal dependencies were investigated via an iterative lag selection procedure. Lagged versions of the target variable representing the demand from the previous 1 to 7 days were sequentially incorporated one at a time, with the model retrained and evaluated after each addition using metrics including R^2, RMSE, and MAE. The 1-day lag consistently demonstrated superior performance across iterations; the same procedure, when applied to the subsequent ensemble models, likewise identified the 1-day lag as optimal, thereby establishing it as the standard lag structure. Random Forest was then introduced as a bagging approach: multiple decision trees are trained on bootstrap resamples and their predictions are averaged, which primarily reduces variance at the cost of increased complexity. The non-lagged forest with 100 estimators and all features ("Random Forest, 100 est.; feats=all") already outperformed both linear baselines, but its almost perfect training scores and clearly weaker test scores pointed to overfitting. Hyperparameter tuning was therefore carried out by constraining tree depth, the minimum number of samples per leaf and per split, and by limiting and subsampling features ("Random Forest, 200 est.; depth=30; leaf=17; split=2; feats=6; samp=0.5"). This tuned forest yielded substantially better test errors and a smaller train–test gap, indicating that controlling model capacity and feature usage effectively mitigated variance while still improving on the linear models.

Building on the lag analysis, the same one-day lag of the target was then incorporated into the Random Forest ("Lagged RF, lag=1d; 200 est.; …"). In both Tables 2 and 3, this specification achieved higher test R^2 and lower RMSE/MAE than the non-lagged forests, with only a moderate discrepancy between training and test metrics. This shows that combining carefully tuned bagging with the most informative lag structure delivers a more favourable bias–variance trade-off and more reliable predictions for both pickups and dropoffs. Finally, Gradient Boosting was applied to the lag-1 feature set ("Gradient Boosting, lag=1d; 200 est.; 6 feats"). This model relies on boosting, where trees are added sequentially so that each new tree is trained to predict the residual errors left by the previous trees. By iteratively correcting these residuals, the ensemble can reduce bias more aggressively, while variance is controlled through the number of boosting stages, the depth of each tree, and the learning rate that scales the contribution of each new tree. Even with a relatively simple hyperparameter choice, the lagged Gradient Boosting models in Tables 2 and 3 achieved the highest test R^2, the lowest RMSE and MAE, and very small train–test gaps. This indicates that, for the chosen clusters and the one-day lag structure derived from the preliminary analysis, boosted trees provide the most effective balance between bias and variance and thus the most reliable next-day, hour-by-hour demand forecasts, while also illustrating that there is no universal best algorithm and performance depends critically on both hyperparameter tuning and the underlying feature design.

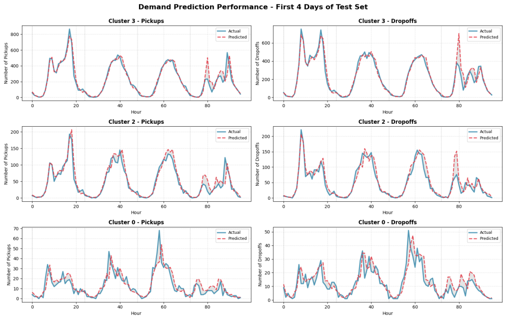

Figure 5. Hourly demand predictions vs. actual values over 96 hours for three representative clusters. Model successfully captures daily patterns and peak-hour dynamics.

3.3. Bike Repositioning Strategy

Accurate demand prediction enables operational planning, but the real challenge is ensuring customers can actually find bikes when they need them. A cluster might have balanced demand over 24 hours, but if pickups surge in the morning and dropoffs peak in the evening, bikes accumulate at the wrong locations throughout the day. We need to reposition bikes overnight to correct these imbalances.

We calculated repositioning requirements by tracking cumulative bike flow hour-by-hour. During morning rush, pickups exceed dropoffs, creating a growing deficit until evening when the pattern reverses. The maximum cumulative deficit indicates how many bikes we need to move overnight to prevent shortages.

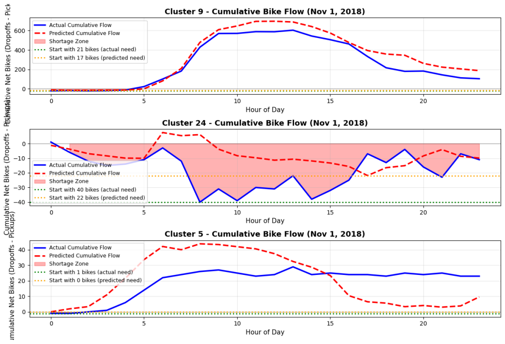

Figure 6. Cumulative bike flow on November 1, 2018. Negative values show where we would run out of bikes without repositioning.

Figure 6 shows what happened on November 1st. Cluster 9 (high-volume commuter hub, 296 avg hourly demand) needed 21 bikes but our model predicted only 17. Cluster 24 (recreational area, 72 avg demand) was more volatile, needing 40 bikes when we predicted just 22. Cluster 5 (low-volume, 5 avg demand) needed 1 bike and we predicted 0 which was essentially correct given the tiny scale.

When we extended this analysis across all test days, we tried adding safety buffers of 10-50% extra bikes beyond our base model predictions. Results varied considerably by cluster. Cluster 9 was the most challenging: even with a 50% buffer, we only achieved 56.7% reliability. The model consistently underestimated this busy commuter station. Clusters 24 and 5 performed better at 85-90% reliability, but with massive over-provisioning (123 and 53 bikes respectively when actual averages were only 27 and 0). Overall, the 50% buffer strategy yielded 75.9% reliability.

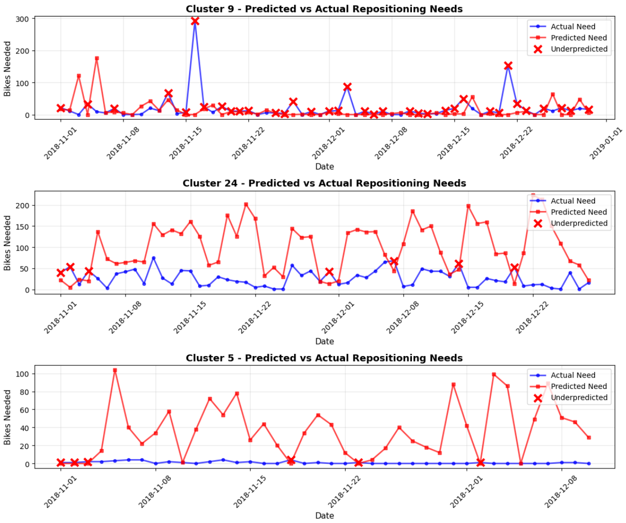

Figure 7. Daily predicted vs actual repositioning needs across the test period (base model without buffer). Red X marks show days where predictions were insufficient.

Figure 7 shows the raw model predictions compared to actual needs throughout the test period. Cluster 9 frequently underpredicts, especially during cold snaps and holidays. Cluster 24's predictions wildly overestimate on most days. This suggests the model handles typical conditions well but struggles with the extreme events that actually matter for avoiding shortages.

3.3.1. Alternative Approach

As a comparison, we also evaluated a simpler approach: using the 90th percentile of historical needs (35 bikes for Cluster 9, 53 for Cluster 24, 3 for Cluster 5). This baseline achieved approximately 90% reliability with reasonable over-provisioning (14-26 bikes), clearly outperforming the model-based approach. This indicates that for daily operations where consistent reliability is essential, historical percentiles might work better than model predictions.

The key takeaway is that while our prediction model is quite accurate for average demand, translating those predictions into reliable operations proved more challenging than expected. Different clusters failed in different ways: Cluster 9 was systematically underestimated, while Clusters 24 and 5 were massively overestimated. The best strategy appears to be using predictive models for long-term planning and fleet sizing, but relying on percentile-based approaches for actual daily repositioning where consistent performance is critical.

4. Exploratory Component: Weather Sensitivity Analysis

Understanding how weather affects bike-sharing demand is critical for operational planning. Section 3 revealed that weather features had very little impact on the predictions, a surprising finding given conventional wisdom about weather's impact on outdoor activities. This exploratory analysis investigates why: does weather sensitivity vary systematically across different types of clusters?

4.1. Research Question and Hypothesis

We hypothesize that cluster type determines weather sensitivity. High-volume clusters serving primarily commuters (who must travel regardless of conditions) should exhibit weather resistance. Low-volume clusters serving recreational riders (who can choose when to ride) should show higher weather sensitivity. If confirmed, this would explain Section 3's finding and suggest that weather-based repositioning strategies are unnecessary for commuter-heavy areas.

4.2. Cluster Characterization

Before testing weather sensitivity, we characterized our three representative clusters (9, 24, 5) using temporal demand patterns. We calculated two indicators: commute ratio (morning and evening peak demand relative to midday) and weekday-to-weekend usage ratio.

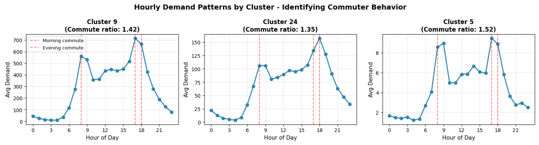

Figure 8. Hourly demand patterns reveal distinct cluster characteristics. Cluster 9 shows strong evening peak (700 trips at 6pm) and high weekday dominance (1.53x), indicating commuter-heavy usage. Cluster 24 shows moderate dual peaks with equal weekday/weekend usage (0.93x), suggesting recreational patterns.

All clusters exhibit some commuter characteristics, but differ in magnitude. Cluster 9 (296.4 avg hourly demand, 23 stations) shows the strongest weekday dominance with pronounced evening peaks and 1.53x weekday ratio. Cluster 24 (72.3 avg demand, 17 stations) displays balanced dual peaks with a unique pattern: weekend usage actually exceeds weekday usage (0.93x ratio). Cluster 5 (4.8 avg demand, 14 stations) shows clear commute patterns despite low volume, with moderate weekday dominance (1.19x).

4.3. Precipitation Impact

We compared average hourly demand on clear days (no precipitation) versus rainy days for each cluster. Contrary to expectations, precipitation showed minimal impact across all cluster types.

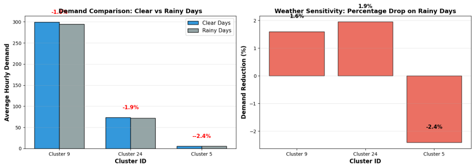

Figure 9. Precipitation impact comparison. Left: absolute demand on clear vs rainy days. Right: percentage reduction. All clusters show less than 2.5% demand drop during rain, indicating high rain-resistance regardless of cluster type.

Cluster 9 showed 1.6% demand reduction on rainy days, Cluster 24 showed 1.9%, and Cluster 5 showed a surprising -2.4% (slight increase). These differences are negligible from an operational standpoint. The finding suggests that NYC bike-share riders travel in rain regardless of trip purpose, likely because trips are short (median 7 minutes) and alternative transportation during rain (taxis, rideshare) is expensive and inconvenient.

4.4. Temperature Impact

Temperature analysis revealed a dramatically different pattern. We binned temperatures into five ranges (freezing to hot) and calculated average demand for each cluster.

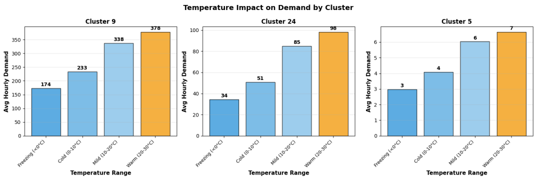

Figure 10. Temperature impact by cluster. All clusters show strong positive relationship with temperature, but differ in magnitude. Demand increases 118-186% from freezing to warm conditions.

Temperature effects were substantial across all clusters. From freezing (<0°C) to warm (20-30°C) conditions, Cluster 9 demand increased 118% (173.5 to 378.2 trips/hour), Cluster 24 increased 186% (34.4 to 98.3), and Cluster 5 increased 123% (3.0 to 6.7). Unlike precipitation, temperature showed clear differences by cluster type: the recreational-leaning cluster exhibited the highest percentage increase.

4.5. Regression Analysis

To quantify weather sensitivity, we fit linear regression models for each cluster: demand $\sim$ temperature + precipitation. This allows us to isolate the marginal effect of each weather variable while controlling for the other. The models achieved modest explanatory power with R^2 values of 0.097 (Cluster 9), 0.162 (Cluster 24), and 0.095 (Cluster 5), indicating that weather explains only 10-16% of demand variance which is consistent with our finding that weather has minimal predictive value for short-term operations.

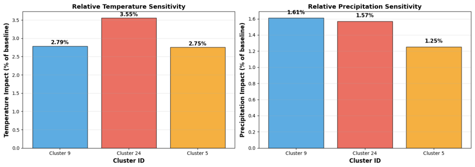

Figure 11. Relative weather sensitivity by cluster (normalized by baseline demand). Temperature shows clear gradient: recreational cluster (24) exhibits 3.55% change per °C, commuter cluster (9) shows 2.79%. Precipitation effects are minimal and uniform across clusters (1.25-1.61% per mm).

When normalized by average demand, the pattern becomes clear: Cluster 24 shows 3.55% relative change per °C (highest sensitivity), Cluster 5 shows 2.75%, and Cluster 9 shows 2.79%, consistent with our hypothesis that recreational clusters are more temperature-sensitive. In contrast, precipitation coefficients showed minimal relative sensitivity (1.25-1.61% per mm) with no meaningful pattern by cluster type, confirming that precipitation does not matter for NYC bike-share demand.

4.6. Discussion

Our analysis supports the hypothesis for temperature but rejects it for precipitation: the recreational cluster shows 27% higher temperature sensitivity (3.55% vs 2.79% per °C), while all clusters are essentially rain-proof with less than 2.5% demand reduction.

However, the practical significance is limited. A 10°C temperature swing would cause a 35.5% demand change for Cluster 24 versus 27.9% for Cluster 9. While this 7.6 percentage point difference aligns with our hypothesis, it is too small to justify cluster-specific operational strategies.

The precipitation finding is more striking. All clusters are essentially rain-proof, with less than 2.5% demand reduction even during active precipitation. This explains why weather features contributed minimally in Section 3's prediction models: the feature simply lacks predictive power for short-term operations.

5. Conclusions

We set out to predict bike-sharing demand and turn those predictions into repositioning decisions. The results are mixed in instructive ways.

On the prediction side, Gradient Boosting with one-day lagged features performed best, achieving test R^2 of 0.806 for pickups and 0.827 for dropoffs. The key was combining the right features (yesterday's demand matters most) with a model that corrects its own errors iteratively. Systematic tuning of learning rates or cross-validation could push performance higher. We only tested three representative clusters spanning high, medium, and low demand, with lower-demand clusters showing weaker accuracy, suggesting prediction models may need cluster-specific tuning for sparser usage patterns.

The repositioning analysis told a more sobering story. Our predictions captured average demand well, but translating them into reliable operations proved challenging. Even with a 50% safety buffer, we only achieved 56.7% reliability for the high-volume commuter cluster. Meanwhile, the 90th percentile of historical demand achieved 90% reliability with reasonable over-provisioning. The takeaway: for daily operations, historical percentiles might beat model predictions. Predictive models are better suited for longer-term planning and fleet sizing.

This study has limitations worth noting. Our spatial clustering groups stations by proximity alone, ignoring land use or transit connectivity. We analyzed only three of forty clusters due to computational constraints, and our temporal features were limited to simple daily lags rather than sophisticated time series decomposition. The dataset covers a single year in one city, so findings may not generalize to cities with more extreme climates or different trip distances.

Future work could address these gaps. Incorporating station-level characteristics (subway proximity, office density, land use) might improve both clustering and prediction accuracy. Testing the approach on multiple cities would reveal whether the ”simple beats complex” finding holds more broadly, or whether it's NYC-specific. Finally, exploring real-time demand updates during the day could salvage model-based repositioning strategies. Our overnight-only approach may be too rigid for continuous rebalancing systems.

The broader lesson is that not all intuitive relationships hold in practice. NYC bike-sharing turns out to be remarkably weather-resistant, likely because trips are short and alternatives are expensive. For operators, the message is clear: invest in predictive models for strategic planning, but stick with simple historical percentiles for daily operations. Focus repositioning strategies on what actually drives demand: recent history, time of day, and day of week, and save the sophisticated weather forecasting for seasonal fleet adjustments.