42577 Introduction to Business Analytics

Andrej Anisic
Mail: s252848@dtu.dk
StudentID: s252848

Vasileios Konstas
Mail: s252879@dtu.dk
StudentID: s252879

Georgios Loulakis
Mail: s252920@dtu.dk
StudentID: s252920

Athanasios Karaiskos
Mail: s253775@dtu.dk
StudentID: s253775

***Abstract*—Report Contribution**

All group members were present during meetings and adhered to the agreed timeline.

- **Vasileios: K-means clustering; Time-series demand prediction model; Graph creation and graph embeddings model development.**
- **Andrej: Data cleaning, preprocessing and visualization; DBSCAN clustering; LSTM prediction model.**
- **Georgios: Mean Shift clustering; exploratory demand–weather analysis.**
- **Athanasios: Day-type pattern exploration ; hourly usage distribution analysis; station activity grouping and cross-day consistency checks.**
- **Joint Tasks: Net-flow prediction (Vasileios & Georgios), LSTM prediction model for graphs (Andrej & Vasileios)**

I. Introduction

The following report presents an analytical study of a bike-sharing dataset with over 17 million logs. The essential data cleaning and pre-processing was applied in the dataset. The analysis employed clustering methods (K-means, Mean-Shift and DBSCAN), finding that K-means (for k = 20) performed best in identifying station communities. The prediction challenge forecasted hourly pickups and dropoffs using SARIMAX and Neural Networks and introduced net-flow calculation for determining bike repositioning needs. Finally, graph embeddings (Node2Vec) were used for clustering based on functional connectivity, identifying central hubs for strategic management. An exploration of usage patterns across weekdays, weekends and holidays revealed distinct bimodal (commuting) and unimodal (leisure) daily profiles. The analysis was extended to link Citi Bike demand to hourly NYC weather conditions (2018), utilizing Python and Meteostat data, feature engineering, and grouped statistics with pre-determined K-Means clusters to quantify the impact of rain, cold, and heat on demand, culminating in scenario tables and a lookup function that provides operational tools for prediction and rebalancing strategies.

II. Data Analysis and Visualisation

The data provided as a CSV file contained 17,548,339 rows (each row represents one bike trip made by one person) and 14 columns (features of each trip). An initial inspection of the data showed that only the columns *start_station_id* and *end_station_id* contained missing values, with 2,497 missing entries in each column, while the other columns had no missing values. Instead of filling the NaN values, all rows with any missing value were removed, which is approximately 0.014% of the data and is not expected to affect the analysis. The column *Unnamed : 0* contained only an index and no useful information, so it was also removed. The string columns *starttime* and *stoptime* were parsed into new datetime columns using the format specified in the data. No exact duplicate rows or logical errors in the dates were found (i.e., stop time before start time).

To ensure that all trips are within the New York area, filtering based on geographical coordinates was applied for both start and end stations. In total, 82 trips with incorrect or out-of-area coordinates and 8 trips that do not belong to the year 2018 were eliminated. Trip duration, originally measured in seconds, was converted into minutes for easier interpretation. The validity of the trip duration values was checked by comparing them to the duration calculated from the start and stop times, and it was found that 151 trips have significantly large errors.

Year of birth of the rider was converted to age and filtered to retain only realistic values (between 10 and 90 years). The age distribution of riders can be seen in Figure 1.

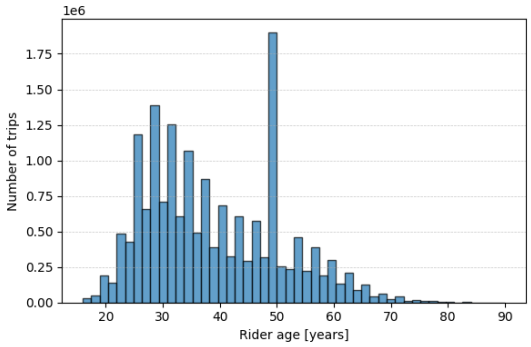

Fig. 1. Age distribution of riders

The gender field was left as encoded in the source data, because a check of unique values showed three categories with a non-negligible amount of unspecified gender (see Appendix Figure 11). The *usertype* feature indicates whether the rider is a long-term subscriber or a casual customer (89.1% vs 10.9%). User type was converted to a binary encoding (*Subscriber*: 1 and *Customer*: 0). Finally, to focus the analysis on typical bike trips, extreme trips were removed (longer than 3 hours), which is about 0.16% of the data (see Figure 2).

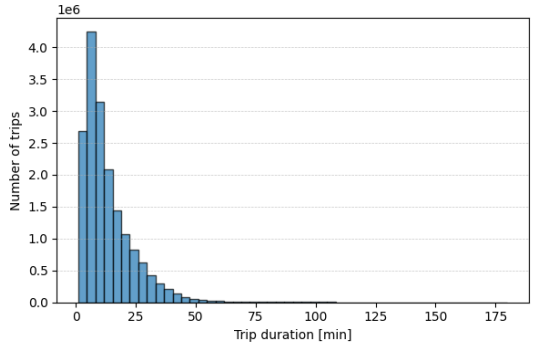

Fig. 2. Trip duration distribution

Figure 3 shows the distribution of daily trip counts for each month in 2018. The number of trips during May-October is much higher compared to the rest of the months. There are also individual days whose total number of trips is unusually different from the other days in the same month.

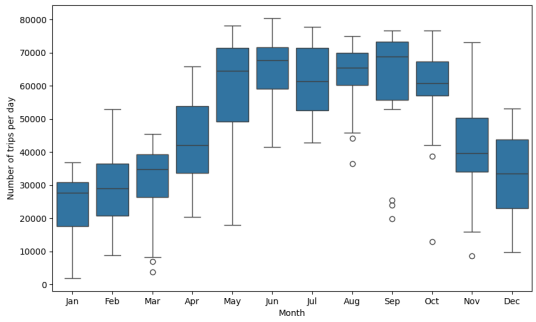

Fig. 3. Distribution of daily trip counts by month

III. Prediction Challenge

Clustering

Three clustering methods were applied to the bike-sharing dataset, each requiring careful parameter tuning to optimize performance. **K-means** primary hyperparameter , the number of clusters **k=20**, was selected using the elbow method and combining visual intuition with the respective silhouette scores. **Mean-shift** clustering identifies cluster centers by shifting candidate centroids towards regions of higher point density. Our tuning process focused on the bandwidth parameter **(bandwith=0.369)**, which controls the size of the search window and influences the granularity of discovered clusters. Lastly, **DBSCAN** required the optimization of two key parameters, the ε-neighborhood radius and the minimum number of points (MinPts) required to form a dense region [2]. We determined $\boldsymbol{\varepsilon}$**-neighborhood=385m** and **MinPts=5** as the best combination resulting in 23 clusters and 115 noise stations. There were difficulties to achieve required number of clusters and to reduce number of noise points. The performance of the three tuned models was evaluated using Silhouette, Davis-Bouldin (DBI) and Calinski-Harabasz (CH) scores.

Metric	Mean-Shift	K-means	DBSCAN
Silhouette	0.3795	0.4084	0.1519
DBI	0.7010	0.7494	0.7063
CH	862.6	1153.7	281.56

TABLE I
CLUSTERING PERFORMANCE COMPARISON

Overall, k-means performs the best, while DBSCAN performs poorly in all metrics. The k-means indicates well-seperated clusters with a silhouette score of **0.4084** for **k=20 clusters.** A real-map visualisation is shown in Figure 12.

Pickups/Dropoffs prediction

The aim of this section is to build a prediction model in order to be able to forecast what the demand of both pickups and dropoffs for a cluster of stations will be over the next 24 hours. We have implemented two models, time series analysis using SARIMAX and LSTM neural network. The aim is to assess how well an SARIMAX baseline can capture daily seasonality in pickups and dropoffs, and then to evaluate whether a LSTM can provide any improvement.

Time Series Analysis: For the time series model, the trip data are aggregated to obtain separate univariate series of hourly pickups and dropoffs for 2018. We make sure to use a continuous hourly index and fill any missing timestamps with zero counts. Then, we plot autocorrelation and partial autocorrelation. These reveal strong daily seasonality at a 24-hour period and short-term dependence at the previous hour, leading to usage of SARIMAX models with first-order differencing and both non-seasonal and seasonal autoregressive terms. The final model uses SARIMAX$(1,1,1) \times (1,1,1)_{24}$. We fit the model using these inputs on January-October data and then evaluate our implementation on November-December data by calculating the root mean squared error (RMSE) and Mean Absolute Error (MAE). Included in the Appendix, Table X shows next day's prediction results for clusters 2,3,15 and Figure 4 and Figure 5 are next day's pickups and dropoffs plot for cluster 2, respectively.

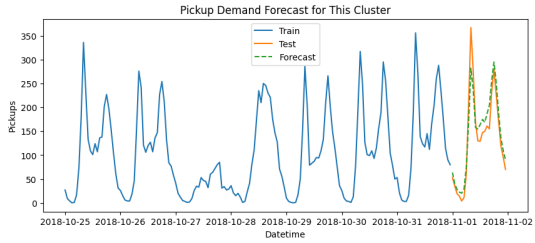

Fig. 4. SARIMAX pickups prediction for 01-11-2018 for cluster 2

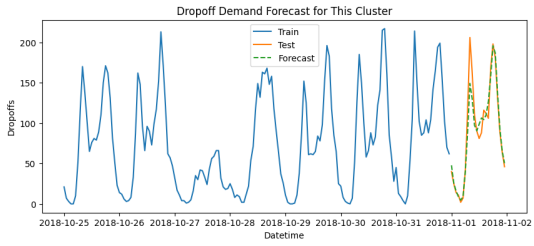

Fig. 5. SARIMAX dropoffs prediction for 01-11-2018 for cluster 2

Neural Networks: A univariate LSTM model was implemented separately for pickups and dropoffs. For each target, every training sample consisted of the previous seven days of hourly demand for the selected cluster (168 time steps of normalized counts), and the network produced the 24 normalized hourly values for the next day. Model architecture was composed of two LSTM layers, followed by a dense layer with ReLU activation and a final linear output layer with 24 neurons [3]. The model was trained using Adam optimizer, with MSE as the loss function. January–October days were used for model development, where the last 30 training days were reserved as a validation set and the remaining earlier days formed the training set, whereas all days in November–December were kept as an unseen test set. A small grid search over the number of LSTM and dense units (32, 64, 128), dropout rate (0.1, 0.3, 0.5), and learning rate (1e-3, 5e-4), combined with early stopping on validation loss (see Figure 6), was used to select the final hyperparameters and number of training epochs. After selection, the model was retrained on the union of training and validation data and evaluated on the test set, with forecasts inverse-transformed back to the original scale before computing RMSE and MAE (see Figure 7 and 8).

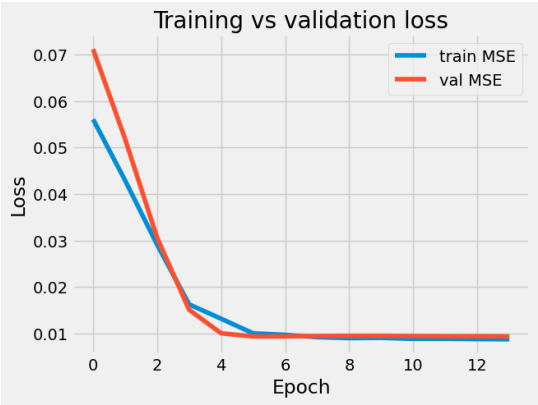

Fig. 6. Training/Validation loss for cluster 3 dropoffs prediction (first set of hyperparameters)

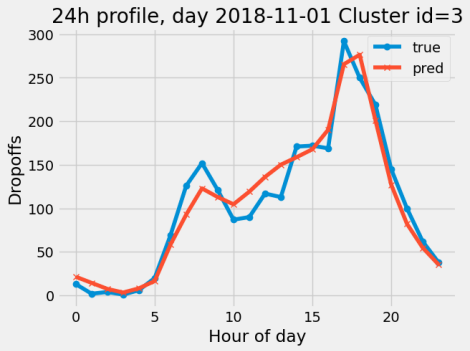

Fig. 7. Dropoffs prediction/ground truth for 01-11-2018 for cluster 3 with LSTM

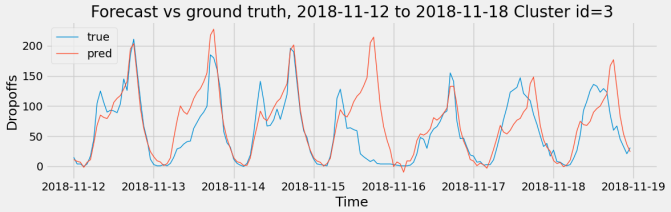

Fig. 8. Dropoffs prediction/ground truth from 01-11-2018 to 2018-11-18 for cluster 3 with LSTM

Model Comparison: Overall, across all three clusters, LSTM performs consistently better than the time series baseline in terms of both pickups and dropoffs. LSTM achieved lower MAE and RMSE on the test set for pickups. The LSTM cuts errors for every cluster in pickups, with the most significant gains coming from cluster 15 since it is the one for which the baseline errors are particularly high. Similarly, for dropoffs, the LSTM achieved substantially lower MAE/RMSE than the baseline in all the clusters. This shows that nonlinear temporal dependencies captured by the LSTM are more appropriate to model hourly bike demand than the simple time series model (see Tables II and III).

TABLE II
PICKUPS PREDICTION COMPARISON FOR CLUSTERS 2,3,15

cluster	Pickups	MAE test set	RMSE test set
2	Time Series	93.19	111.11
	LSTM	29.39	42.83
3	Time Series	38.91	56.41
	LSTM	28.97	43.29
15	Time Series	287.66	323.91
	LSTM	80.18	121.85

TABLE III
DROPOFFS PREDICTION COMPARISON FOR CLUSTERS 2,3,15

cluster	Dropoffs	MAE test set	RMSE test set
2	Time Series	33.98	47.36
	LSTM	33.3	49.33
3	Time Series	13.34	18.74
	LSTM	26.69	40.94
15	Time Series	105.67	120.18
	LSTM	67.37	103.89

Net-flow calculation

Once again, we use hourly pickups and dropoffs which are aggregated for each cluster we pick. Therefore, we have continuous time series of incoming and outgoing trips per cluster per day per hour. Since we have already implemented time series analysis once in Task II, we try to make the workflow more flexible. Train/test split is fully automated. The user specifies a target data and the code dynamically constructs a training window before that date and a test window from that day onwards. Hourly pickups and dropoffs are forecasted for the defined day and so, the equation that calculates hourly netflow is

$$\text{net\_flow}_t = \text{pickups}_t - \text{dropoffs}_t$$

. Finally, the number of bikes that need to be repositioned during the night in order to cover targeted day's demands is equal to the maximum hourhly net flow. The following table shows the number of bikes to be repositioned for clusters 2,3 and 15.

Cluster	No. of bikes
2	133
3	196
15	624

TABLE IV
NUMBER OF BIKES TO BE REPOSITIONED ON 2018-11-01 PER CLUSTER

IV. EXPLORATORY COMPONENT

How do graph-based station communities and central hubs improve the operational and strategic management of the bike-sharing network?

Graph embeddings map each station (node) in the bike-sharing network to a vector in a low-dimensional space so that nodes with similar connection patterns end up close together in that space. Instead of using raw longitude and latitude, which only encode physical distance, embeddings capture how stations are actually used together (who rides where, how often, and via which "corridors"), so they preserve functional similarity and community structure that geography alone cannot represent. This makes them more informative inputs for clustering and downstream models, because stations that are far apart but serve similar flows can still be grouped, while nearby stations with very different connectivity are distinguished.

In this project, an undirected graph for the whole of 2018 is built where each station is a node and an edge exists between two stations if at least one trip occurred between them; a trip from A to B is treated the same as from B to A, and the edge weight equals the total number of trips between the pair. Degree distributions reveal that most nodes are highly connected. Node2Vec [1] is then used to generate 64-dimensional embeddings. The resulting vectors are clustered by choosing $k = 16$ combining the elbow method and the silhouette score of 0.155. We use PCA[13] to reduce the embedding dimensions for visualization but as the silhouette score suggests, we do not observe clear separation between the clusters. Then, we work on finding the most central nodes in each cluster; nodes that work as "bridges" between clusters. This is achieved by calculating degree and betweeness centrality for every station. Finally, a real-map visualization is created. Central stations for every cluster are highlighted.

Fig. 9. Real-map cluster visualization for k=16 clusters based on graph embeddings

Graph-based clustering reveals operational communities that can **define more realistic service zones** for rebalancing and routing trucks, because stations within the same graph cluster exchange many trips and therefore benefit from being managed as a unit when setting target inventories or planning redistribution tours. Second, the identified central stations with high betweenness highlight a small set of **critical hubs** where disruptions or stockouts would affect multiple communities, so operators can **prioritize them for capacity upgrades**, real-time monitoring, and dynamic rebalancing resources. In Figure 9, critical hubs align with major traffic hubs such as the Central Park, bridges (Williamsburg, Queensboro, Manhattan,Brooklyn), and are close to port or waterfront transfer areas. Third, the clusters and their hubs can **guide**

network expansion and marketing by indicating where additional docks, new stations, or tailored pricing campaigns are likely to capture demand spilling over between highly connected areas of the city.

Finally, we perform pickups and dropoffs prediction for three of the most central clusters using the Neural Networks model from the prediction challenge. Table V shows the MAE and RMSE calculated for each clusters prediction. In general, the model seems to make better predictions regarding pickups than dropoffs. Pickups and dropoffs next day plots can be found in the Appendix[14][15].

TABLE V
NEURAL NETWORKS PERFORMANCE

cluster		MAE		RMSE	
		next day	test set	next day	test set
1	Pickups	82.52	119.82	122.94	179.68
	Dropoffs	120.03	144.97	168.8	204.57
4	Pickups	46.10	82.88	68.84	124.08
	Dropoffs	54.75	94.99	79.43	139.71
9	Pickups	**31.94**	56.67	**52.64**	83.99
	Dropoffs	**32.28**	56.38	**47.05**	84.62

Extended research using weather data

Weather is a key external driver of bike demand. Here, CitiBike usage is linked with hourly weather to see how temperature and rain shift demand beyond the usual hour–of–day and weekday/weekend patterns.

Hourly 2018 weather for Central Park (temperature, precipitation, wind, pressure) is cleaned and converted to consistent units. Trips are aggregated to hourly pickups and merged on the timestamp, giving a single *hourly demand + weather* dataset used in the following analysis. Out of 8 761 hourly records, only 351 (4%) lack precipitation values, while all other weather variables are fully observed. To examine how weather relates to system-level demand, we combine hourly CitiBike pickups with Meteostat [4] temperature data and apply a 7-day moving average to both. Figure 10 shows that normalized demand tracks temperature closely: both increase from winter to a summer peak and then decline, indicating a clear seasonal link between warmth and bike usage.

Figure 20 summarizes the average change in hourly bike demand under two non-baseline weather scenarios. Demand is about 40% lower on cold rainy weekdays (temperature below 5°C with rain) and nearly 80% higher on warm dry weekends (temperature above 25°C without rain) compared to the dry 10–15°C weekday baseline.

We evaluate the time–weather regression (a log–linear multiple regression with calendar and weather covariates, following standard practice in [5, Ch. 3]) by comparing predicted and observed hourly pickups on the test set. The extended model with temperature and rain improves test R^2

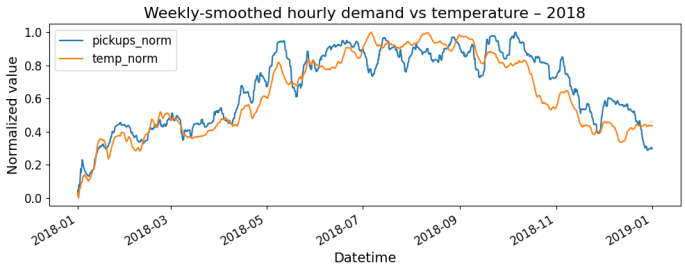

Fig. 10. Weekly-smoothed hourly CitiBike demand (normalized pickups) and normalized air temperature in 2018. Both series share a similar seasonal pattern, with higher demand in warmer months and lower demand in winter.

compared to the time-only baseline, using a simple train/test split on 2018 data [6, Ch. 4], and Figures 21 and 22 show that it tracks daily and weekly patterns well, with larger errors mainly at the highest peaks.

In the last step, we link weather to the spatial clusters and turn it into simple operational rules. We aggregate hourly pickups by station cluster and merge them with the hourly weather data, so that each record contains (cluster, time, weather, demand). For every cluster we estimate a mild-weather baseline and average demand under rain, cold (<5°C) and heat (>25°C), and derive simple rain/heat multipliers that capture its weather sensitivity. Figure 25 summarizes these effects across clusters, while Figures 23 and 24 illustrate the full response patterns for two representative cases.

Combining CitiBike usage with external weather data reveals strong and systematic effects of temperature and precipitation on demand, both at system and cluster level. The resulting scenario multipliers and cluster-level lookup tables allow planners to reduce idle bikes and unnecessary truck movements on low-demand days, while better matching capacity to peak demand in warm, dry conditions. In practice, this improves service quality for riders and helps lower operating costs, increasing the revenue generated per deployed bike.

Weekdays/Weekends Pattern Exploration

In this section, we investigate the patterns associated with different day types, including weekdays, weekends, and holidays. To achieve this, we classified each day into one of these categories. We then plotted the total daily trips against the dates and applied a 7-day moving average to visualize the patterns. The trend lines are color-coded: red for holidays, blue for weekdays, and green for weekends. We observe that bicycle usage is most frequent on weekdays, likely due to commuting. Usage decreases on weekends and is lowest on holidays, though we note that the data for holidays is limited as shown in Figure 18 in the Appendix. We then analyzed the average trip duration per category to assess potential leisure usage during free time. We observed that the average trip duration is shorter on weekdays compared

to weekends and holidays (Table VI). This indicates that while total usage volume is lower on non-working days, cycling is likely used for recreation, as evidenced by the longer duration of individual trips. We can see this in the

Category	Average Duration (in s)
Weekday	926.50
Weekend	1170.30
Holiday	1091.28

TABLE VI
AVERAGE TRIP DURATION BY CATEGORY.

following graph 16 that displays the daily average trip duration for each category, plotted alongside a 7-day moving average. We then investigated potential trends regarding user age across these categories. The results are presented in the following Table VII and indicate that there is no correlation between age and the type of day. We subsequently

Category	Average Age	Median Age
Weekday	39.15	37.00
Weekend	38.60	36.00
Holiday	38.86	36.00

TABLE VII
AVERAGE AND MEDIAN USER AGE BY CATEGORY (IN YEARS).

analyzed gender trends within each category. We observe that the proportion of female users remains consistent across all categories. Conversely, the proportion of male users is comparable between weekends and holidays but is notably higher on weekdays. We hypothesize that this variation is linked to account registration status: regular weekday users are likely registered subscribers who have specified their gender, whereas weekend and holiday traffic may consist of more non-subscribers or casual users who have not provided this information. The results are shown in Table VIII.

Category	Unknown (0)	Male (1)	Female (2)
Holiday	12.27%	64.19%	23.54%
Weekday	6.51%	70.66%	22.83%
Weekend	14.13%	61.06%	24.81%

TABLE VIII
GENDER PERCENTAGES BY CATEGORY.

This trend is corroborated by the results below (Table IX, which demonstrate that the proportion of non-subscribed users almost triples on holidays and weekends compared to weekdays. Then we proceeded with the analysis of the

Category	Non-subscriber	Subscriber
Holiday	17.35%	82.65%
Weekday	7.86%	92.14%
Weekend	20.20%	79.80%

TABLE IX
USER TYPE PERCENTAGES BY CATEGORY.

average daily departures and arrivals for each station across each category. To aid in future demand categorization, the stations were grouped into four clusters based on usage intensity: Low, Moderate, High, and Very High Activity. We observed that the relative busyness of stations is largely independent of the day type; stations that are busiest on weekdays generally remain the busiest on weekends and holidays. While absolute usage volumes fluctuate, the relative hierarchy of station demand remains consistent (Figure 17).

Finally, we plotted the average number of trips per hour for each category to investigate daily usage patterns. The results verify our hypothesis regarding distinct usage behaviors. Weekdays exhibit a clear bimodal distribution with sharp peaks at 08:00 and 17:00, consistent with commuting dynamics to and from work. In contrast, weekends and holidays display a unimodal, Gaussian-like distribution that peaks during daylight hours, indicating a prevalence of leisure-oriented usage spread throughout the day (Figure 19).

V. CONCLUSIONS

This project showed how combining rigorous data preparation, spatial clustering, time series modeling, and graph-based analysis can turn raw Citi Bike logs into concrete operational insights for a large bike-sharing system. K-means clustering with 20 geographically coherent station groups provided a strong basis for cluster-level forecasting, while LSTM models substantially reduced hourly prediction errors compared to a SARIMAX baseline, with MAE drops of more than 60% for pickups in some clusters (for example, from 287.66 to 80.18 in cluster 15). Net-flow based calculations, then, translated these forecasts into actionable overnight repositioning targets, indicating, for instance, that up to 624 bikes would need to be rebalanced in a single cluster to avoid shortages. The exploratory components further enriched these results: Node2Vec graph embeddings revealed functionally central hubs and realistic service communities, and the integration of hourly weather data quantified demand shifts of roughly 40% under cold rain and +80% in warm, dry weekend conditions relative to a mild weekday baseline, leading to simple scenario multipliers and lookup tools for planners. Together, these elements demonstrate that advanced analytics can significantly enhance both predictive accuracy and decision support for bike-sharing operations, improving service reliability while enabling more efficient and adaptive fleet management.

REFERENCES

[1] node2vec: Scalable Feature Learning for Networks. A. Grover, J. Leskovec. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2016

[2] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pp. 226–231, 1996.

[3] TensorFlow, "tf.keras.layers.LSTM," TensorFlow API Documentation, https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM, accessed Dec. 5, 2025.

[4] Meteostat Python Library. Retrieved from https://meteostat.net/en/. Accessed: December 2025.

[5] James, G., Witten, D., Hastie, T., & Tibshirani, R. *An Introduction to Statistical Learning: With Applications in R* (2nd ed.). Springer, 2021. Relevant material: Chapter 3 (Multiple Linear Regression).

[6] Géron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.). O'Reilly Media, 2019. Relevant material: Chapter 4 (Training Linear Models with Scikit-Learn).

APPENDIX

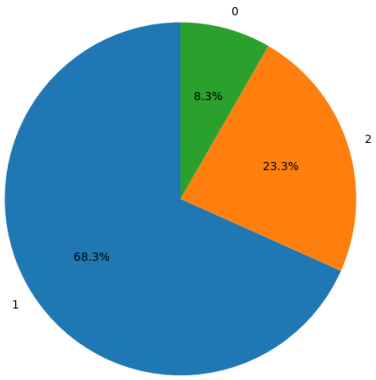

Fig. 11. Trips ratio by gender

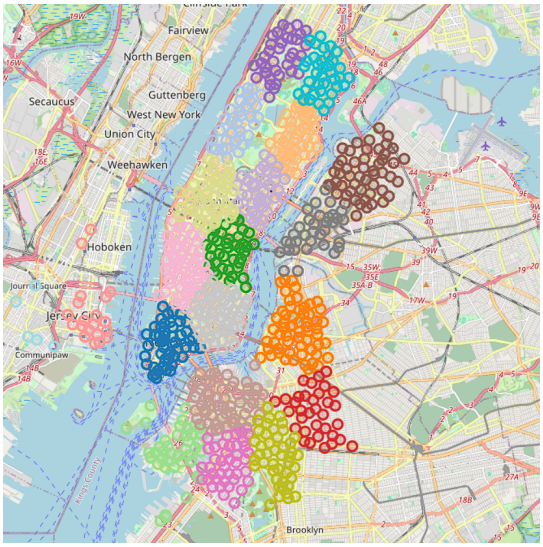

Fig. 12. Real-map cluster visualization for k=20 clusters

TABLE X
TIME SERIES PREDICTIONS FOR 1-11-2018 CLUSTERS 2,3,15

cluster	Prediction	MAE next day	RMSE next day
2	Pickups	23.78	29.12
	Dropoffs	10.57	16.80
3	Pickups	14.32	23.08
	Dropoffs	8.426	11.07
15	Pickups	57.27	64.04
	Dropoffs	25.05	29.03

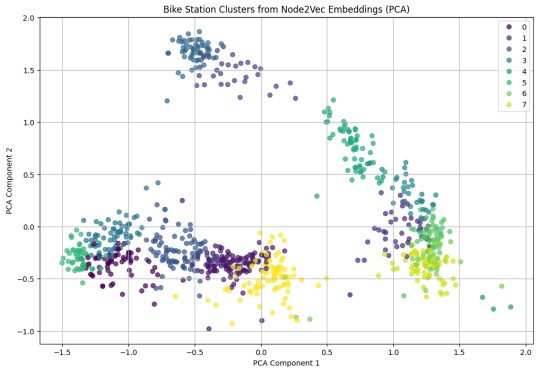

Fig. 13. PCA output of 64-dimension embedding vectors for k=16 clusters

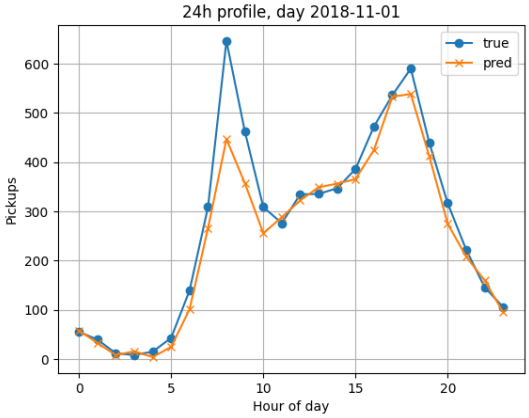

Fig. 14. Pickups prediction for 01-11-2018 for cluster 9 based on graph-embeddings based k-means clustering with k=16

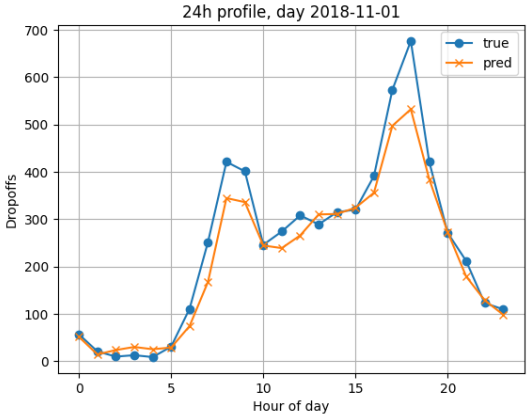

Fig. 15. Dropoffs prediction for 01-11-2018 for cluster 9 based on graph-embeddings based k-means clustering with k=16

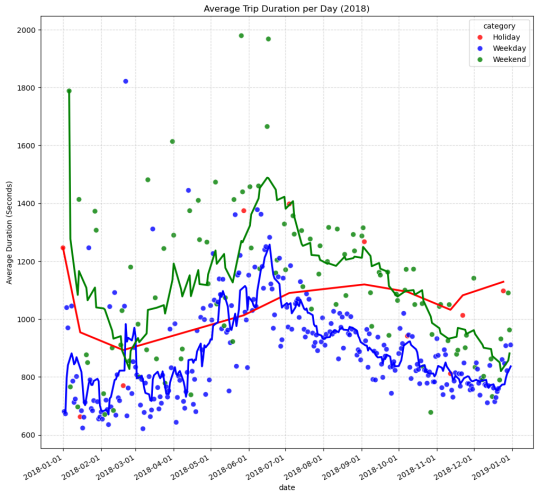

Fig. 16. Average trip duration per day in seconds for 2018.

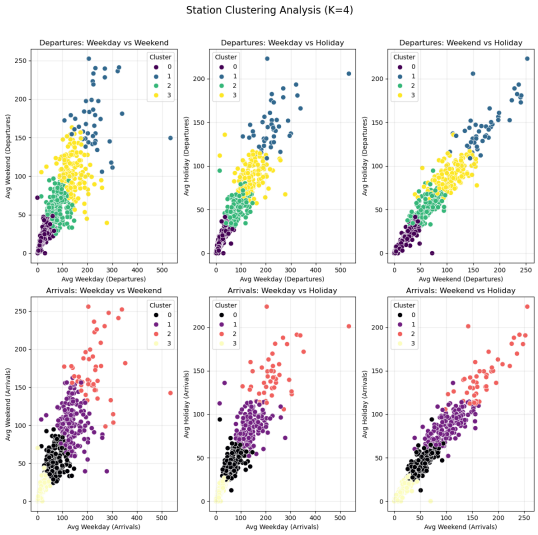

Fig. 17. Station clustering according to usage for all day categories against each other.

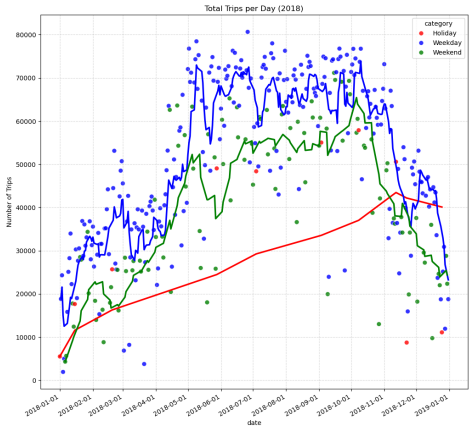

Fig. 18. Total trips per day in absolute numbers for 2018.

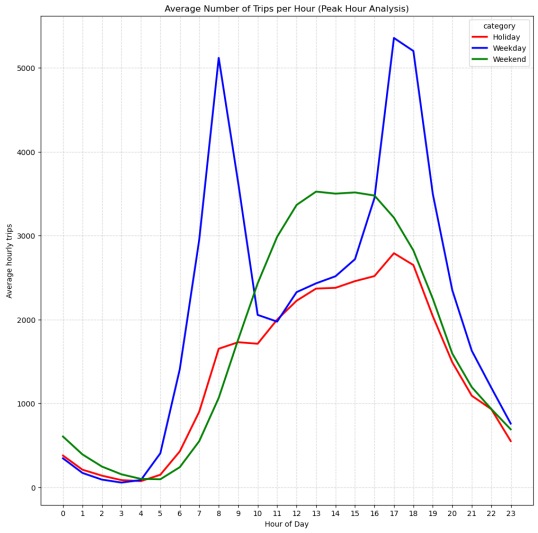

Fig. 19. Average number of trips per hour for each category for 2018.

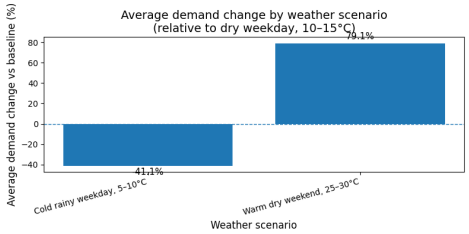

Fig. 20. Average demand change by weather scenario (relative to a dry weekday, 10–15°C).

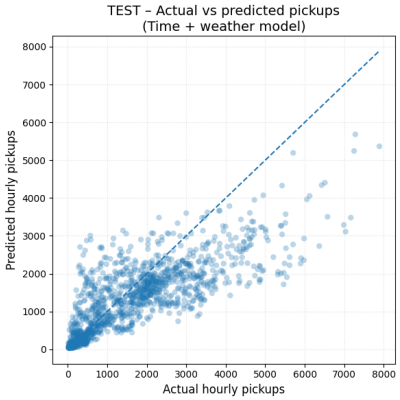

Fig. 21. Test set – actual vs predicted hourly pickups for the time+weather model.

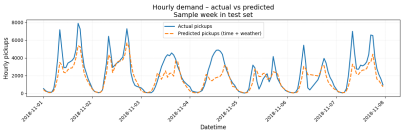

Fig. 22. Sample test week – hourly demand compared with predictions from the time+weather model.

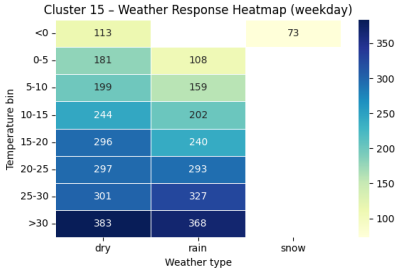

Fig. 23. Cluster 15 – expected hourly pickups by temperature bin and weather type (weekday).

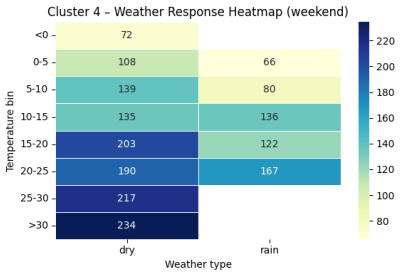

Fig. 24. Cluster 4 – expected hourly pickups by temperature bin and weather type (weekend).

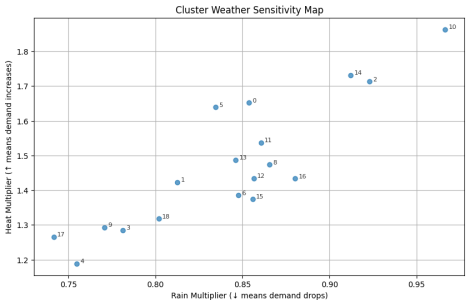

Fig. 25. Cluster-level weather sensitivity map: rain multiplier (horizontal axis) versus heat multiplier (vertical axis), with one point per cluster.