Robust Machine Learning Framework for Bike-Sharing Demand Prediction

Hachem Squalli El Houssaini, Ilyass El-ogri, Yassine Ouali, Hiba Chougrad
Département Génie des Systèmes Intelligents
ENSA Fès, Université Sidi Mohamed Ben Abdellah
Fès, Morocco

***Abstract*—Bike sharing systems are so important for healthy environment and green transportation, so understanding the patterns and manage demand of bike for both users and company that fix them is necessary, for this reason we use the Capital Bikeshare data from Washington, D.C as basis to create our machine learning model predict the hourly bike demand, our work focus on three to manage this problem: firstly data preprocessing like eliminating the pre-pandamic era, secondly adding feature engineering by adding new values to strengthening the correlation or founding new relation as (lag feature and temperature x hour). Finally, we solved our problem by choosing our final model and parameter based on cross-validation of time-series. CatBoost achieved the best performance, resulting in an R^2 of 0.9595, an RMSE of 111.27 bikes/hour, and an MAE of 68.51 bikes/hour. In addition, the chosen features showed the most impact on the model, as lag features are the best and interactions are the last. Our Method surpasses the company's simple model by minimizing error by 34%.**

***Index Terms*—Bike-sharing, demand forecasting, time series analysis, feature engineering, time series, urban mobility**

I. Introduction

A. Context and Motivation

Urban transportation becomes the way to go to save the environment by reducing carbon emissions, improving traffic flow, reducing the use of petrol vehicles, and not forgetting to make transportation more accessible for everyone. We see the increase in Bike-sharing systems (BSS) around the world, with more than 3000 systems operating as of last year [1]. If we take Washington, D.C., Capital Bikeshare as an example, we find that they have 3.5+ million rides annually at 700+ stations [2]. So it is necessary to match the demand with the supply for the progress of BSS.

It is not an easy task because the factors controlling the system are diverse, dynamic, and many types of data, ranging from meteorological to spatial data.

The problem of demand imbalance: the company suffer as we say previously a demand and supply problem across all city where company cannot predict the precise time where large locations suffer a shortage (limiting rentals) or Excesses, which makes the company lose money and service credibility [3], because of moving bikes with trucks a count as 15 to 20% of operating costs [4]. This makes rebalancing demand a must to reduce cost and improve service quality.

B. Research Challenges

Difficulties in Predicting Bike-Sharing Demand:

Concept Drift: the sudden change of behavior. Between 2020 and 2021, COVID-19 **was a time event in the century, changing people's behaviour from a socially active entity to being more isolated. Almost everyone works remotely during lockdowns, so using this temporary data will create a biased model based on** one-off behaviors.

Complex link between weather and space (High-Dimensional Spatio-Temporal Dependencies): Demand changes based on:

- *Time (Temporal patterns):* like rush hours between (8-9 AM, 5-7 PM) start and end work time , also weekdays and holidays should be considered with seasons.
- *Short-Term Influence (Lag dependencies):*is Logical to recognize the most important information is indeed the history itself knowing demand from previous hour or week because of strong autocorrelation (demand at time t predicts t+1).
- *Weather:* Temperature and precipitation can control the choice of using a bike [6].

Issues with data quality: the dataset contains many outliers (e.g., wind speed reported as zero), missing values, or measurement errors. This should be cleaned and processed.

C. Research Objectives and Contributions

This paper presents a machine learning pipeline for predicting hourly BSS demand in the city. Our contributions include:

1) *Rigorous Data Cleaning (Preprocessing):*

- **Excluding data from** 2020-2021 to avoid "concept drift" and just letting data from 2022-2024 with 2.1 million trips.
- **Outlier fixing:** Wind speed zeros were replaced with the 1st percentile (6 km/h), and also the highest speed stayed at 35 km/h (see Equation 3 in Methods).

2) *Smart Feature Creation:*

- **Temporal:** adapting hour and month for cyclical nature [7] using sin/cos math as hot encoders.
- **Lag Features:** including all previous hours from 1 hour, 24 hours, and 168 hours ago, additionally, the means (3h, 12h, 24h windows).
- **Combined Features:** Mixing Weather with Time (e.g., temp $\times$ hour, rain $\times$ rush_hour).

3) *Model Testing and Best Setup:*

- **Tested Five Models:** Benchmarking Linear Regression, Random Forest, XGBoost, LightGBM, and CatBoost.
- **Time-Series Validation:** Using(TimeSeriesSplit, 5 folds) for cross-validation to keep the data in chronological order and prevent data leakage.

4) *Understanding the Model (Ablation):*

- **Measuring Impact:** Quantifying the marginal contribution of each feature group (Lag, Weather, Time) to prediction.
- **Feature Importance:** Analysis reveals which features contributed most to prediction (e.g., demand from 1 hour ago (lag_1h) accounted for 36%).

D. Paper Organization

The paper is organized as follows: Section II defines the problem, Section III explains the methodology of each part, including preprocessing, feature engineering, and models. **Section IV** presents results, ablation studies, and comparisons. **Section VI** discusses the limitations of our approach.**Section VII** concludes the paper with future directions.

II. Formulation of the Predictive Problem

A. Notations and Task Definition

Let $y_t \in \mathbb{N}$ represent the total number of bike rentals recorded at hour t. We also define the feature vector $\mathbf{x}_t \in \mathbb{R}^d$ capturing the usual mix of temporal indicators, weather conditions, and a small collection of autoregressive inputs. Given the observed data set

$$\mathcal{D} = \{(\mathbf{x}_t, y_t)\}_{t=1}^{T},$$

our goal is to train a non-linear regression function f with parameters θ. In practice, this function yields the one-step-ahead prediction

$$\hat{y}_{t+1} = f(\mathbf{x}_{t+1}; \theta),$$

which we tune to minimize its forward error on future observations.

B. Objective Function

Because y_t is count-based and generally right-skewed, we apply a variance-stabilizing transformation using $\log(1 + y_t)$. The training objective follows the Root Mean Squared Logarithmic Error (RMSLE), mainly because it responds more naturally to multiplicative discrepancies:

$$\mathcal{L}(\theta) = \sqrt{\frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x},y)\in\mathcal{D}} \left(\log(1 + y) - \log(1 + \hat{y})\right)^2}.$$

C. Performance Validation Metrics

To evaluate how well the model generalizes, we convert predictions back to the original count domain using the inverse mapping

$$y = e^{\hat{y}_{\log}} - 1.$$

The standard metrics used throughout the relevant literature are considered here:

- Root Mean Squared Error (RMSE),
- Mean Absolute Error (MAE),
- Coefficient of Determination (R^2),
- Mean Absolute Percentage Error (MAPE).

III. Materials and Methods

A. Data Acquisition and Preprocessing

The study utilizes historical Capital Bikeshare trip records (Washington D.C.) merged with daily meteorological observations. The raw event-based logs $E = \{e_1, ..., e_N\}$ are aggregated into a discrete time series y_t representing the total rental count at hour t :

$$T_t = \{\text{trip} \mid \text{trip} \in \text{all\_trips}, \lfloor \text{trip.ts} \rfloor_h = t\} \quad (1)$$

$$y_t = |T_t| \quad (2)$$

where τ_i is the trip timestamp and $\lfloor \cdot \rfloor_h$ denotes flooring to the nearest hour a standard approach for demand forecasting [11]. This hourly resolution ($\mu \approx 179$, $\sigma \approx 171$) balances signal fidelity with operational utility.

1) Concept Drift Mitigation: The COVID-19 pandemic (2020-2021) introduced significant non-stationary **concept drift**, where the demand distribution $P(y_t|\mathbf{X}_t)$ diverged substantially from standard urban mobility patterns. To ensure the model learns generalized behavior and avoids temporary biases, we restricted the dataset to the post-recovery regime: $\mathcal{D}_{\text{train}} \subset \{t \mid t \geq \text{Jan 1, 2022}\}$. Post-filtering analysis confirms the restoration of canonical bimodal commute peaks (Pearson $\rho(y_t, y_{t-24}) = 0.83$).

2) Physical Consistency & Noise: Sensor noise and inconsistencies were corrected to ensure physical stability. Anomalous zero-value wind records ($N = 1313$) were imputed using the 1^{st} percentile of non-zero values ($W_{p1} \approx 6\,\text{km/h}$), which serves as a minimum expected wind speed. High-end outliers were subsequently capped at the 99^{th} percentile to mitigate sensor error influence. Small time-series gaps ($< 8\%$) were filled via linear time-interpolation to preserve temporal continuity as shown in [12].

B. Feature Engineering

Washington DC Bikeshare System has some complex dynamics that is affected by temporal and spatiotemporal dynamics therefor we transformed the timestamps and the meteorogical data into a high dimension feature space ,after cleaning the data and removing the pandemic related outliers from 2019 to 2022 , our encoding produced 36 variables , that we can categorize into 5 specific domains: Temporal and Cyclic Domain, Astronomical and Physical Domain , Societal Domain, Meteorological domain and Autoregressive Domain

*1) **Temporal and cyclic features**:* Given the nature of dataset , the count of bikes used In the city can vary considerably between hours ,differentiating high usage from low usage periods, which means that this temporal data hold a categorical type of nature , same thing goes for the other temporal However representing 24 hours a day and 7 days a week and for weeks a month and 12 months a year into

a categorical features will take a lot of space and memory , from 4 features (hour/day/week/month) to 48 distinct categorical features And to address that we will be using Cyclical Encoding ,it represents the relation between timestamps giving the fact that 23:00 is close to 0:00 However the Euclidean distance remain high despite there closeness in time

$$d(23, 0) = |23 - 0| = 23$$

We will transform our datetime values into numerical data with sin and cosine using the idea of unit circle to represent the cycling nature of our data , so instead of mapping all the values into feature we get only per (day ,week,..) hour_sin,hour_cos, month_cos,month_sin,daw_sin,daw_cos

$t_{\sin} = \sin\left(\frac{2\pi t}{T}\right)$, $t_{\cos} = \cos\left(\frac{2\pi t}{T}\right)$ applied on (hour, day, week, month)

now the eucledian distance in this vector space [tsin,tcos] becomes consecutive ,small correctly presenting the proximity of time $||\mathbf{v}_{23} - \mathbf{v}_0||_2 \approx ||\mathbf{v}_t - \mathbf{v}_{t+1}||_2$

And also we have kept the original features to allow the tree based models to perform splits (hour, month, day_of_week) [14]

*2) **Physics based environmental modeling**:* We have added solar kinematics : solar_declination , and hour_angle to get solar_elevation (angle which shows that it has better correlation than temperature $R_\alpha \sim 2 \times R_{\text{temp}}$ logically daylight acts like a positive factor on visibility and safety for riders , meanwhile the rise of temp makes the people uncomfortable

Solar Declination (δ): The angle between the rays of the sun and the plane of the earth's equator.

$$\delta \approx 23.44^\circ \times \sin\left(\frac{360^\circ}{365} \times (N + 284)\right)$$

[15]

(N : the day of the year)

Hour Angle (h):$h = 15^\circ \times (LST - 12)$ (LST : the Local Solar Time)

Solar Elevation Angle (α):

$$\sin(\alpha) = \sin(\phi)\sin(\delta) + \cos(\phi)\cos(\delta)\cos(h)$$

(ϕ : the latitude of Washington DC) [16]

Physiological stress apparent_temp using Australian Apparent Temperature (AT) to account for the cooling affect of wind and humidity on cyclists Also the feature day of year to represent change due to seasonality and other unknown factors to the city Physiological Stress (Australian Apparent Temperature):We use the linear approximation for Apparent Temperature (AT) to account for wind chill and humidity:

$$AT = T_a + 0.33e - 0.70v - 4.00$$

T_a: dry-bulb temperature (°C), e: water vapor pressure, v: wind speed (m/s) [17]

*3) **Societal and behavioral constraints**:* Added is_holiday using the Us federal holiday calendar and is_weekend is_working day ,which helps identify the change of behavior during certain periods of time for the users ,additionally is_rush_hour that was engineered between 07:00-09:00 and 16:00-19:00

Rush Hour Indicator:

$$I_{\text{rush}}(t) = \begin{cases} 1 & \text{if } t \in \cup \\ 0 & \text{otherwise} \end{cases}$$

Holiday/Weekend Logic:

$$I_{\text{workday}} = \neg(I_{\text{holiday}} \vee I_{\text{weekend}})$$

*4) **Meteorological thresholds and interactions**:* Weather inputs (temp,humidity,precip,windspeed,cloudcover) , and we added boolean inputs like Is_raining , Is_snowing , Is_bad_weather which plays a huge role in deciding the leisure conditions for a given time .

$I_{\text{is\_raining}} = (P > 0.1)$,
$I_{\text{is\_snowing}} = (P > 0.1) \vee (T < 2)$,
$I_{\text{bad\_weather}} = (P > 0.1) \vee (W_s > 35)$

Additionally composite effect hour_x_temp for measuring heat effect during certain hours of the day , temp_x_weekend to measure the temperature effect during weekends from workdays , rain_x_rushour to see how rain affect the rush hour traffic ,humidity_x_temp which is highly significant to identify the nature of the weather and it does correlate greatly with peoples outdoor activity patterns

Composite Interaction Features: To capture the joint effect of two variables x_1 and x_2 (e.g., hour x temp):

$$x_{\text{interaction}} = x_1 \times x_2$$

[18]

*5) **Autoregressive features**:* We have introduced lag features , which set the structure of our work to make this model behave in temporal dependency making it a timeseries model ,bringing lag for 1,24,168 hours to capture recent trends, and predict real time anomalies which outperforms static features

Meaning the system will heavily depend on the latest results , and rolling mean for 3,12,24 hours which tracks patterns in data to smooth fluctuations ,noises and understand gradual shifts from unexpected events And to prevent data leakage , the computing started of the first week data lag_168h which made it necessary to remove the first week from the dataset .

Meanwhile the lag features improves the model prediction capabilities , but the constraint is the inability to predict more than a single hour to the future , to fix this , we train the model with and without the lag_1h and rolling_mean_3h , to predict the next 24h using lag_24h or keep only the lag_168h to make the whole next week of prediction available , sacrificing a bit of accuracy for practicality ,that is why for the training , we are going to discuss an ablation part for the role of different features to identify the significance of each singular feature and its role and its practicality in decision making and efficient use of the model . Lag Features:For the target variable y at time t, the lag feature of order k is:

$$L_k(y_t) = y_{t-k}$$

(Used for $k \in \{1, 24, 168\}$ hours)Rolling Mean Features:To smooth noise and capture trends over a window w:

$$\mu_w(t) = \frac{1}{w} \sum_{i=1}^{w} y_{t-i}$$

(Used for $w \in \{3, 12, 24\}$ hours) [19]

C. Algorithm methodology

To predict the hourly bikeshare demand [10], we benchmarked five different regression algorithms - from parametric linear to From models to advanced implementations of gradient boosting.

1) Baseline and Tree-Based Ensembles: We used Linear Apply regression as a baseline to establish relationships, which are linear between the engineered features and the target variable. As Interpretable, this parametric approach basically assumes a linear additive relationship that can normally not capture complex nonlinear demand patterns linked to both weather and temporal factors.

shifts.

We have used Random Forest to handle non-linearity. Ensemble method using bagging (Bootstrap Aggregating). The prediction is averaged from several de-correlated decision trees, Random Forest reduces variance and improves Generalisability compared to single decision trees.

We furthered our exploration into boosting frameworks with XGBoost, Extreme Gradient Boosting, and LightGBM. Both algorithms are based on Gradient Boosting Decision Trees. GBDT, which builds trees greedily to correct the errors of previous members of the ensemble. LightGBM, in It features optimizations such as Gradient-based One-Side Sampling (GOSS) to deal with large datasets.

Well.

2) CatBoost (Categorical Boosting): The main model Of particular interest for this study is CatBoost [9], a GBDT framework. The following section introduces new methods of tackling prediction shift and categorical features.

3) Ordered Boosting: Standard GBDT algorithms face That is a problem of prediction shift: the distribution of $F(x_k)|x_k$ for a training example x_k shifts from the distribution of $F(x)|x$ for a test example x. CatBoost solves This using Ordered Boosting.

Let be a random permutation of the training dataset. Consider each example x_i, a separate model M_i is trained using only the examples which come before x_i in the permutation σ. The residual This can be calculated as: for the i-th sample in the t-th iteration:

$$r_i^{(t)} = y_i - M_{\{\sigma(j)<\sigma(i)\}}(x_i) \tag{4}$$

This ensures that the target value of the current object is not used in computing its own prediction gradient, thereby We obtain an unbiased estimate of the gradient and improve generalization on unseen data.

4) Oblivious Trees: Unlike XGBoost or LightGBM, which usually grow asymmetric trees either level-wise or leaf-wise. CatBoost builds Oblivious Trees (symmetric trees). In the oblivious tree, the same splitting feature and threshold are used across an entire level of the tree.

More formally, a tree of depth d partitions the feature space into 2^d regions. Each of the regions is defined by a binary vector, by the sequence of splits. The prediction value for a leaf index L is computed as:

$$\hat{y}_{leaf} = \sum_{j=1}^{J} W_j \cdot I\{x \in R_j\} \tag{5}$$

This symmetric structure is a strong regularizer that prevents overfitting and allows very fast inference; This is because the tree index can be computed using efficient bitwise operations.

IV. Experiments and Results

A. Data Preprocessing Pipeline Validation

To verify the effect of our prediction quality, we use some methodology to conduct an ablation study, as seen in (Fig. 3). Four steps were chosen to evaluate the chronological test (2024 T3-T4), using the Catboot model as standard with these parameters (500 iterations, depth=6).

Step 1: Unfiltered Raw Data: In the beginning, the models were trained on the whole data from 2020 to 2024; as a result, the model was unstable with RMSE = 212,78 biks/h and $R^2 = 0.47$. Also, the residual analysis (Fig 4) shows a red scatter more widely scattered than the blue cloud, resulting in a big dispersion of error standard deviation by 168bikes/h.

Step 2: Correction of physical outliers: After removing Data of 1.04 million representing the pandemic era, we get RMSE = 212,78 biks/h an improvement of -1.25%, resulting in a restoration of temporal autocorrelation $\rho(y_t, y_{t-24}) = 0.83$.

Step 3: Handling the physical outliers: Correcting the 1,313 sensors' impossibility of getting zero no wind speed by replacing it with the first quartile ($P_1 = 6\,\mathrm{km/h}$) and letting the cap as ($P_{99} = 35\,\mathrm{km/h}$), therefore a cumulative improvement of -3.60%, RMSE = 205.13 bikes/h.

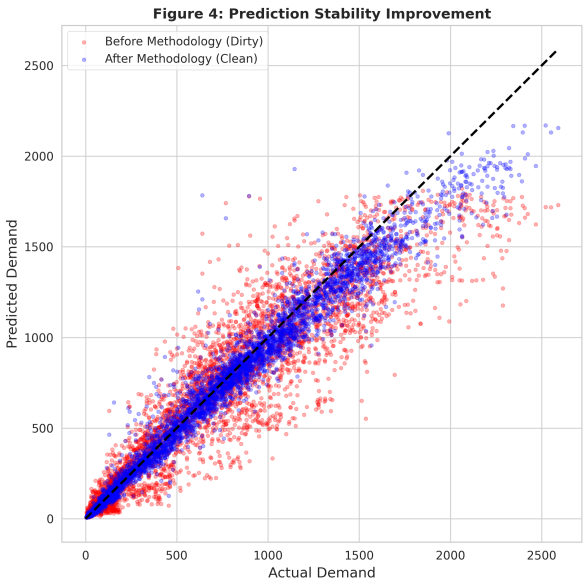

Fig. 1. RMSE Reduction via Data preprocessing Stages

Step 4: Complete the Pipeline of Data Cleaning: After including all features that will be discussed in the next subchapter of feature engineer results, we have RMSE = 109.40 bikes/h, $R^2 = 0.89$, a huge improvement from the baseline. So applying advanced engineering without preprocessing has a bad effect on results, as we see in Step 1, data yielded RMSE = 198.6 bikes/h.

TABLE I
DATA QUALITY IMPACT ON RMSE REDUCTION

Methodology Stage	RMSE	Improv. (%)
Raw Data (No Cleaning)	212.78	0.00
+ Concept Drift Filter (2022+)	210.13	1.25
+ Physics Correction (Wind)	205.13	3.60
Final Proposed Pipeline	**109.40**	**48.59**

B. Correlation analysis

from the results provided from the table 2 we conclude that there is an Autoregression dominance with lag features ,consistently with time series nature of our system And the dominance of lag_168h(r=0.93) over lag_24h(r=0.91) proves that the system reflect strong weekly seasonality , mirroring the days of the week in a strong sense , and the rolling_mean_3h(r=0.77) proved to provide high filtering of the noise

The solar_elevation (r=0.61) shows a higher correlation than hour(r=0.59) which validate our interpretation of the importance of daylight in this problem And the The solar_declination (r=0.23) captures seasonality better than month(r=0.086)

And generally the combined features (eg: temp_x_hour) provides better results than the raw features) So basically our feature engineering provided us with a strong correlation with our target count_log

Feature	Correlation
count_log	1.000000
lag_168h	0.929054
lag_1h	0.922879
lag_24h	0.907973
rolling_mean_3h	0.768976
solar_elevation	0.608924
is_daylight	0.595787
hour	0.591037
hour_angle	0.591037
temp_x_hour	0.537463

TABLE II
FEATURE CORRELATIONS WITH TARGET VARIABLE

C. Model Comparison

The models were evaluated based on the Root Mean Squared Error. RMSE, Mean Absolute Error MAE, and the Coefficient of Determination (R^2) on the chronologically separated test set representing the last 20% of the dataset.

Table presents the performance metrics for all five models. CatBoost outperformed in all respects metrics, achieving the lowest RMSE and highest explanatory Power.

TABLE III
BENCHMARKING RESULTS ON TEST SET

Model	R2 Score	RMSE	MAE
LR	0.9146	161.69	105.77
Random F	0.9375	138.29	85.84
XGBoost	0.9456	128.98	79.63
LightGBM	0.9508	122.75	74.66
CatBoost	0.9595	111.27	68.51

The linear baseline worked well, with R2 = 0.91. but significantly outperformed by tree-based ensembles. Among boosting algorithms, CatBoost lowered the RMSE by 27.8% compared with the Linear Regression baseline and 5.7% compared to LightGBM.

1) Cross-Validation Analysis: First, to validate stability, we performed 5-fold Time Series Cross-Validation. CatBoost maintained low error rates and Variance across folds.

- Random Forest : CV RMSE = 0.2321 0.0174 (log scale)
- XGBoost: CV RMSE = 0.2259 0.0226 (log scale)
- LightGBM : CV RMSE = 0.2190 0.0175 (log scale)
- CatBoost: CV RMSE = 0.2190 0.0168 (log scale)

V. DISCUSSION

A. Interpretation of Data Quality

As shown, the ablation study highlights the importance of data quality, not just that the feature engineer and algorithm parametric choices are enough to have a good performance, as seen in the Figure 1 improvement, nearly half, 48.59% of the final model.

Why? The decision of removing (2020-2021) was impactful to capture the amplitude of current demand in a way that shows how the traffic was historically very low because of lockdowns, so the model learn a diluted average as highlighted in Figure 1, effectively losing the information to detect the modern rush hours consequence of concept drift, making it under-predict for high-demand hours. Furthermore, also shifting meteorological data represents more the real experience because the forecast for the same day is unavailable during morning in our metrics, being inflated by $\sim 18\%$ and also R^2 drops from 0.67 to 0.49 in initial tests, so giving the model the same constraint as the users of bikes creates a false impression of real-world performance.

B. Limitations

Additionally, we think our dataset of the bike-sharing system is missing out a critical component because there are many factors that control the demand of bikes as the recent studies indicate the association of other riding transport as they act as substitute or reverse as we can take example if subways are full or delays people tends to choose other methods of mobility [13] thus excluding this features e.g., subway status, ride-hailing pricing), our model will struggle with demand spike as we seen before in final figure of models caused by the public transport breakdowns a critical factor for urban planning.

C. CatBoost Performance

The benchmarking experiment demonstrated a clear hierarchy of model capability, with gradient boosting methods significantly performing traditional approaches. CatBoost was the best performing model, yielding an R^2 of 0.9595. The superiority of CatBoost can be attributed to to its Ordered Boosting technique. For time-series forecasting, maintaining distributional consistency between training and testing is paramount. First, calculate residuals using only preceding data points in its permutations, CatBoost obtains an unbiased estimate of the gradient has lower bias in The final ensemble

Furthermore, the usage of Oblivious Trees provided a stronger regularization. Given that high dimensionality was introduced by our lag features, such as lag 168h, standard trees in XGBoost might overfit to particular noise in the training history. CatBoost's symmetric structure forced the model to learn more generalized split conditions that applied across the entire dataset, improving its ability to extrapolate to the test set.

1) Feature importance and dynamics: The feature importance analysis shows that the problem Is fundamentally nonlinear but highly autoregressive.

- Lag Features: The dominance of lag 1h, which is 43% importance) and lag 24h shows that recent history is the strongest predictor, confirming the autoregressive nature of the system.
- Weakness of Linear Models: Linear Regression struggled (MAPE 18.87%) because it could not capture the conditional dependencies between weather and tim for for example, the fact that rain dampens demand significantly more during rush hour, at 5 PM, than at 3 AM.
- Robustness: Though LightGBM was at a close second, CatBoost provided the best balance between accuracy and consistency, thereby making it the best option for deployment in a Production forecasting system.

VI. Conclusion

This paper introduces a solid machine learning framework designed to predict hourly bike sharing demand in Washington, DC. Our findings highlight the importance of thorough data preprocessing; by addressing "concept drift" from the pandemic and correcting physical outliers, CatBoost emerged as the top performer among the algorithms we tested, achieving an R-squared of 0.9595 and an RMSE of 111.27 bikes per hour. The model effectively navigates the complex non linear relationships between weather conditions and time-based patterns, with autoregressive lag features proving to be the key predictors. Right now, this framework serves as a practical tool for operators, helping them rebalance their fleets proactively to lower operational costs and enhance service reliability. Looking ahead, we plan to explore the integration of real -time multimodal data, like subway status, to boost the model's adaptability to sudden changes in urban mobility and to strengthen the system's resilience against external shocks.

References

[1] Shaheen, S., et al. (2020). "Evolution of bike-sharing systems: A global perspective." Transport Reviews, 40(3), 287-310.

[2] Capital Bikeshare Annual Report (2024). Washington D.C. Department of Transportation.

[3] Raviv, T., et al. (2013). "Static repositioning in bike-sharing systems: Models and solution approaches." EURO Journal on Transportation and Logistics, 2(3), 187-229.

[4] O'Mahony, E., & Shmoys, D. B. (2015). "Data analysis and optimization for (Citi)Bike sharing." AAAI Conference on Artificial Intelligence.

[5] Teixeira, J. F., & Lopes, M. (2020). "The link between bike sharing and subway use during the COVID-19 pandemic." Transportation Research Part A, 135, 343-353.

[6] Gebhart, K., & Noland, R. B. (2014). "The impact of weather conditions on bikeshare trips in Washington, DC." Transportation, 41(6), 1205-1225.

[7] Zhao, Y., et al. (2023). "BGM: Demand prediction for expanding bike-sharing systems with dynamic graph modeling." IJCAI, 10008-10016.

[8] Box, G. E. P., & Cox, D. R. (1964). "An analysis of transformations." Journal of the Royal Statistical Society, 26(2), 211-252.

[9] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). "CatBoost: unbiased boosting with categorical features." Advances in Neural Information Processing Systems, 31.

[10] Fanaee-T, H., & Gama, J. (2014). "Event labeling combining ensemble detectors and background knowledge." Progress in Artificial Intelligence, 2(2), 113-127.

[11] Jelic, A. (2021). "Predicting bike sharing demand with machine learning." Bachelor's Thesis, Tilburg University.

[12] Jiang, W. (2025). "Prediction of Demand for Shared Bicycles Based on Machine Learning." Academic Journal of Science and Technology, 16(1), 51-60.

[13] Liang, Y., Huang, G., & Zhao, Z. (2022). "Bike Sharing Demand Prediction based on Knowledge Sharing across Modes: A Graph-based Deep Learning Approach." arXiv preprint arXiv:2203.10961.

[14] H. Pelletier, "Cyclical Encoding: An Alternative to One-Hot Encoding for Time Series Features," *Towards Data Science*, May 3, 2024. [Online]. Available: https://towardsdatascience.com/cyclical-encoding-an-alternative-to-one-hot-encoding-for-time-series-features-4db46248ebba.

[15] P. I. Cooper, "The absorption of radiation in solar stills," *Solar Energy*, vol. 12, no. 3, pp. 333–346, 1969.

[16] J. A. Duffie and W. A. Beckman, *Solar Engineering of Thermal Processes*, 4th ed. Hoboken, NJ, USA: John Wiley & Sons, 2013.

[17] R. G. Steadman, "Norms of apparent temperature in Australia," *Australian Meteorological Magazine*, vol. 43, pp. 1–16, 1994.

[18] P. Liu, Z. Pan, Z. Fan, and X. Wang, "The Impact of Weather on Shared Bikes," *Applied Sciences*, vol. 15, no. 17, p. 9834, 2025.

[19] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 2nd ed. Melbourne, Australia: OTexts, 2018.