Data Overview

The dataset comprises records from 30 cyclist counters in Paris, spanning from September 1st, 2020, to September 9th, 2021, with 496,827 observations and 12 features, including numerical variables (e.g., site_id, bike_count), categorical variables (e.g., counter_name, site_name), and datetime variables. The initial dataset is clean, containing no duplicates or missing values, but additional features were added for improved predictive capabilities, necessitating a reassessment of data quality.

Exploratory Data Analysis

1 Season, Working Day, Holiday, Weather

Our analysis indicates higher bike usage during Summer and Fall, particularly on working days and during rush hours. Notably, there's little variation in bike counts based on weather conditions, except during snowfall. This consistency may be attributed to Parisians' preference for biking, even in frequent rain, likely due to commuting needs *(Fig 1)*.

2 Lockdown Dates

During the Covid-19 pandemic, lockdown measures impacted bike counts, especially between October 2020 and June 2021. Stricter rules, including mask requirements, correlated with decreased bike counts in public spaces *(Fig 2)*.

3 Hour

Daily bike count patterns differ on working and non-working days. On working days, peaks occur during morning and evening rush hours, suggesting local commuters. On non-working days, a steady increase is observed throughout the day, likely representing casual bikers, potentially tourists, enjoying favorable weather conditions *(Fig 3)*.

4 Temperature

Analysis of average bike counts in relation to temperature reveals a positive correlation, with higher temperatures associated with increased bike usage. This aligns with the intuitive expectation that warmer weather encourages biking *(Fig 4)*.

5 Correlation Analysis

Analyzing our correlation matrix heatmap *(Fig 5)* reveals significant insights into relationships between numerical variables. Notably:

- log_bike_count exhibits a strong positive correlation with temperature, indicating a preference for biking in warmer weather.
- Conversely, there's a strong negative correlation with humidity, suggesting fewer bike rides during more humid conditions.
- log_bike_count correlates positively with hour, emphasizing the substantial impact of time, especially during rush hours, on recorded bike counts.

Feature Engineering

Following our exploratory data analysis, we refined our dataset to improve the predictive accuracy of our machine learning model. The raw data underwent feature engineering: some features were modified, others removed, and new ones introduced, as detailed below:

- **Temporal Feature Extraction:** The 'date' column which contained the date-time stamp was split into individual year, month, day and hour categorical columns. This breakdown aids in understanding traffic patterns across different time features. We also added additional features such as weekday, season, holiday and working day to capture more trends.

- **Integrating External Data:** We enhanced our dataset with essential weather parameters, such as temperature, humidity, and wind speed, using the meteostat library. This integration, facilitated by the latitude and longitude coordinates of each site, significantly improved our model's predictive capabilities. Additionally, we incorporated pandemic-related data from the lockdowndates library to factor in the effects of Covid-19 lockdowns and mask mandates on cycling activity. This enriched our model with insights into the varied impacts of weather and health safety measures on Parisian bike usage.

- **Transforming and Creating Features:** The dataset was augmented with new categorical variables reflecting key aspects of bike traffic patterns in Paris, including rush hour periods and a weather classification system that considered variations from average precipitation, snow, and wind speed. We also introduced time-of-day categories to delineate morning, afternoon, evening, and night periods. These enhancements were designed to offer a detailed and nuanced view of the influences on bike traffic for our predictive modeling.

- **Final Data Cleaning:** Following the integration of external data sources, we executed a final sweep to clean the dataset, addressing missing values—substituting zeroes for absent rainfall data and applying a KNN algorithm to estimate missing snow measurements. Recognizing the inherent subjectivity in handling missing data, we opted for these practical solutions. We also pruned superfluous columns that offered little predictive value or were redundant. The resulting dataset featured 15 key variables—5 numerical and 10 categorical. To facilitate efficiency in ongoing work, we preserved the cleansed dataset, ensuring that these preprocessing steps are preserved for future analysis and model development.

- **OneHotEncoding:** For constructing our machine learning model, we needed to efficiently handle numerous categorical features. Initial attempts using scikit-learn's methods were computationally intensive. We opted instead for pandas 'get_dummies' function, which streamlined the process. This method converted categorical variables into dummy/indicator variables, from which we removed the original columns and dropped the last dummy variable to avoid collinearity, thus preparing our categorical data more effectively for model training.

- **Potential extensions:** Our project has made significant strides, yet there's room for further refinement. Enriching our dataset with more historical records could surpass the incremental gains from complex feature engineering or hyperparameter optimization. Investigating the impact of local events like the Tour de France and urban factors such as public transit usage, strikes, and construction disruptions could provide comprehensive insights into cycling trends. Model ensembling could also be explored as a way to combine the strengths of individual predictive models, potentially offering a more accurate and robust solution. While

such enhancements were beyond the scope of our initial project due to time and resource constraints, they represent promising avenues for future research to amplify our model's accuracy and reliability.

Modelling

1 Initial Model Benchmarking

We embarked on our analysis by evaluating a selection of regression models, each renowned for its distinct advantages in various data scenarios:

- **Linear models** (Linear Regression, Ridge, Lasso): Chosen for their simplicity and efficiency, these models serve as a baseline for performance.
- **K-Nearest Neighbors** (KNN): Included for its non-parametric, instance-based approach, providing insights into local data patterns.
- **Tree-based models** (Decision Tree, Random Forest, XGBoost, CatBoost, LightGBM): Selected for their ability to capture nonlinear relationships and interactions among features.

This diverse suite was benchmarked using default settings to assess adaptability to the data quickly. While not definitive, this approach efficiently narrows down the field to models demonstrating a promising fit, sidestepping the impracticality of exhaustive tuning for each model.

2 Candidate Model Selection

The selection was grounded in RMSE, leading to the exclusion of models with subpar performance:

- Linear models were discarded due to inadequate handling of complex data patterns.
- The Decision Tree was prone to overfitting, as evidenced by its performance discrepancy between training and testing.

The remaining models—Random Forest, XGBoost, CatBoost, and LightGBM—exhibited a desirable blend of accuracy and generalizability, meriting further tuning.

3 Hyperparameter Tuning Strategy

Our structured tuning strategy comprised:

- **Grid Search with Cross-Validation:** To methodically navigate the hyperparameter space while ensuring robustness against data variability.
- **Randomized Search:** For a computationally tractable exploration of a broader hyperparameter landscape.
- **Optuna Optimization:** Leveraging advanced Bayesian optimization to identify promising hyperparameters more efficiently.

This approach was favored over exhaustive search methods due to its strategic balance between thoroughness and computational intensity.

4 Final Model Optimization and Selection

Integrating the final tuning phase, LightGBM surfaced as the top-performing model, characterized by its swift execution and adeptness at deciphering intricate data patterns—ideal for predicting Parisian bike traffic. Anticipating the Kaggle test set, we expect the optimized LightGBM to reflect our validation's RMSE, a testament to our rigorous and discerning model development process.

From Complexity to Clarity

Our Kaggle journey took off with an overly complex model that, while sophisticated, unfortunately led to a high initial score of 2.1887 due to overfitting. A critical reassessment was in order. We took a step back to re-evaluate our approach, realizing the necessity of simplicity as the cornerstone of a robust predictive model. In response, we pruned the feature set, focusing on the predictive power of each variable. The key to this transformative process was in distinguishing signal from noise. We discarded features such as 'site_name', 'site_id', 'coordinates', and others that cluttered the model without contributing to its predictive strength. This refined approach was instrumental in achieving a notable breakthrough, with the score plummeting to a more competitive 0.6722. We leaned into a minimalistic yet effective preprocessing pipeline, employing standardization and one-hot encoding where it mattered most. We honed in on date components, counter names, and a selection of time-based features that were more reflective of the underlying patterns in urban bike traffic. As we transitioned from XGBoost to CatBoost, and eventually to LightGBM, our model retained its newfound agility, balancing sophistication with simplicity. The inclusion of domain-specific features like school holidays and weather conditions was now strategic, avoiding the overfitting traps of our initial attempts. This meticulous recalibration was validated by our steadily improving scores, culminating in a final public score of 0.6048 and a private score of 0.5900. Our Kaggle evolution—from an initial complex construct to a streamlined, efficient model—underscores the critical importance of iterative modeling, rigorous evaluation, and strategic adaptation. It is a compelling testament to the adage: less can be more, especially when each less is chosen for its measurable impact on model performance.

Insights and Reflections for Future Data Science Projects

Our Kaggle journey culminated in a model that distilled complexity into precision, offering robust predictions of Parisian cycling trends. The strategic paring down from an initial score of 2.1887 to an impressive 0.5900 on the private leaderboard demonstrates the potency of our refined approach.

In future data science endeavors, we will harness this experience to craft models that balance sophistication with practicality, ensuring each feature we include earns its place through measurable performance impact. Our methodology will continue to evolve, always prioritizing the integrity and interpretability of our predictions. This project has sharpened our analytical acumen, preparing us to tackle new challenges with a blend of caution and creativity. The insights gained here will be the bedrock upon which we build, not just for the next competition, but for all our future data science explorations.

Appendix

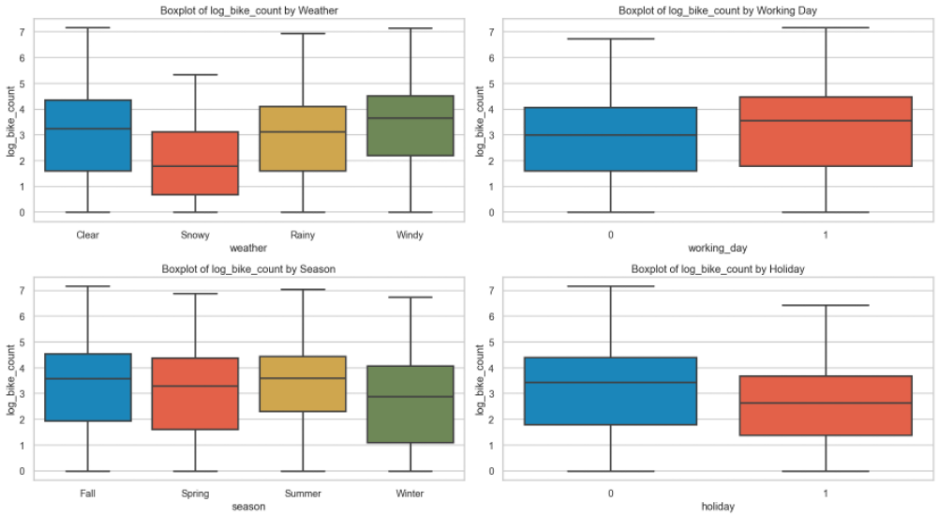

Figure 1: Box plots of log_bike_count across Categorical Variables

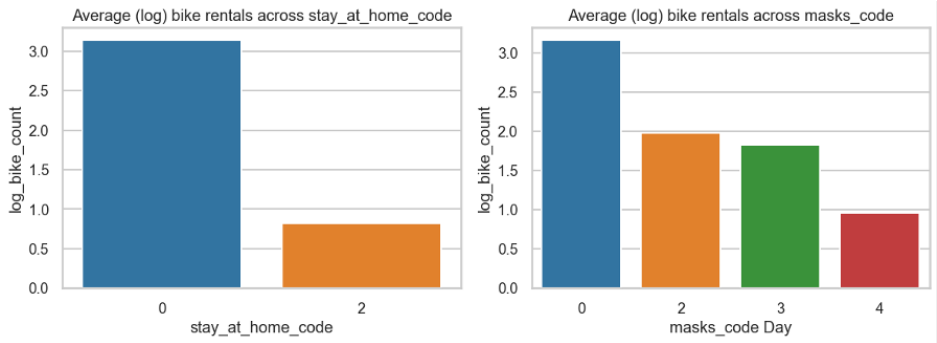

Figure 2: Average bike rentals under Covid-19 restrictions

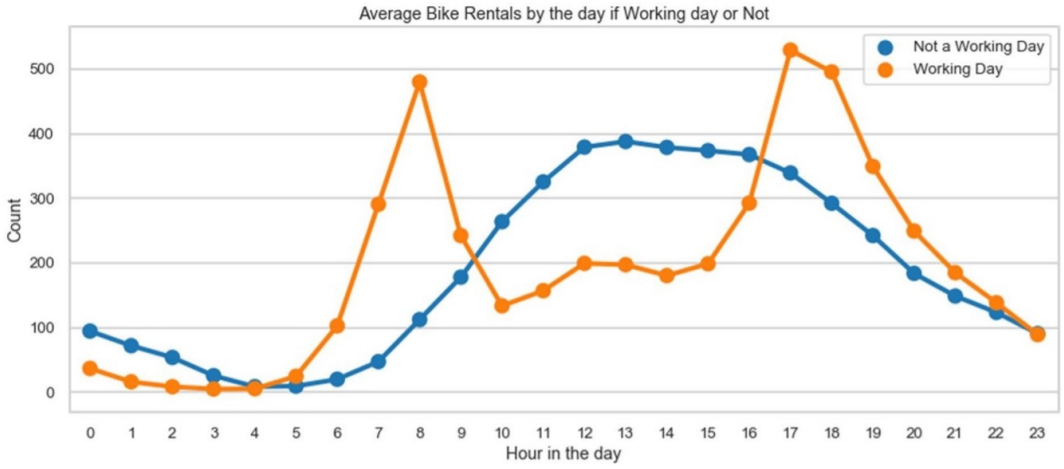

Figure 3: Hourly average bike rentals for Working day or Non-Working day

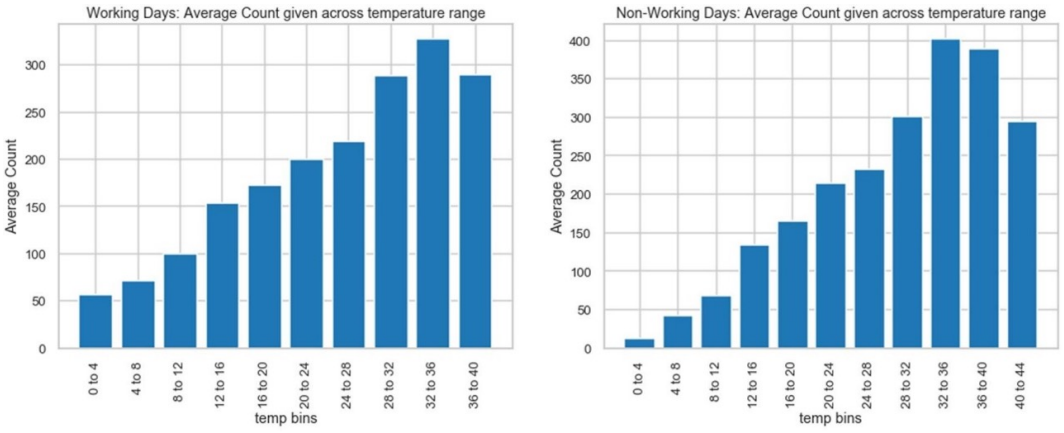

Figure 4: Average bike rentals across different Temperatures

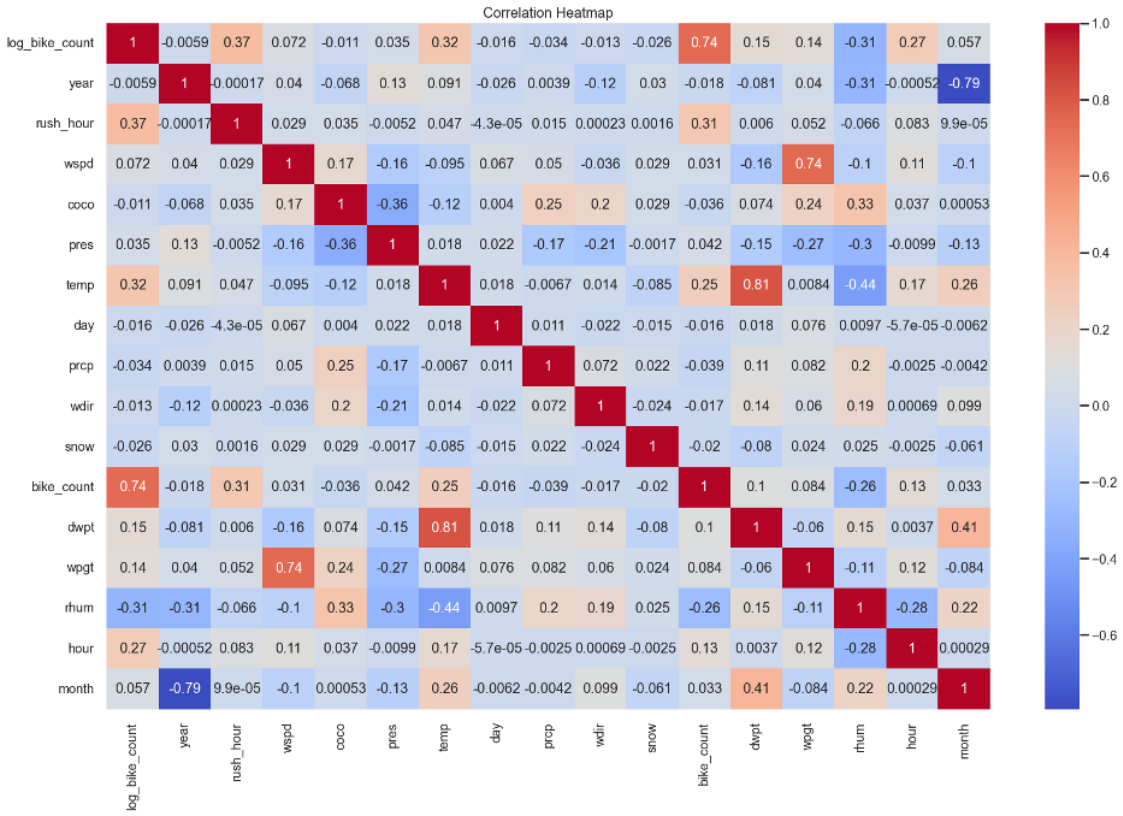

Figure 5: Heatmap of the correlation between all the numerical features