Kaggle Challenge Report: Predicting cyclist traffic in Paris

Jialong Xu, Houcine Hassani Idrissi
École polytechnique
(Dated: December 17, 2023)

This report analyzes cyclist traffic in Paris using a Kaggle dataset. We applied feature engineering and machine learning to uncover influencing factors and predict cycling patterns. Key findings reveal temporal and spatial trends, though correlations with weather were less pronounced. Our insights offer guidance for urban planning and sustainable transport initiatives.

INTRODUCTION

This report delves into understanding and predicting cyclist traffic in Paris, a key aspect of sustainable urban planning. Leveraging a comprehensive dataset from Kaggle, we examined cycling activity patterns using various machine learning methods. XGBoost emerged as our chosen method after rigorous testing and optimization, including feature engineering and parameter tuning through GridSearch. Our analysis focuses on temporal trends and spatial distribution of cyclist traffic, identifying peak times and areas of high activity. Although our attempt to correlate weather data with cycling patterns was less effective, we explore potential reasons for this in the report. The insights gained aim to inform city planners and policymakers, offering a detailed look at the effectiveness of different analytical approaches and their implications for enhancing urban cycling infrastructure.

METHODOLOGY AND DISCUSSION

I. Data Loading and Preprocessing

In this project, we only utilized the data from Kaggle (training data, final test data and external weather data). We began by loading the dataset using Pandas, a powerful Python library for data manipulation and analysis. The dataset includes training data (‘train.parquet‘), test data (‘final_test.parquet‘), and external data (‘external_data.csv‘).

Preliminary preprocessing involved cleaning the data by removing duplicates, handling missing values, and ensuring data quality, consistency and accuracy. We used $df.isnull().sum()$ to find all the null values. We dropped columns that contain too many (more than 90%) nan values, and switch other nan values into 0 or the mean value of the corresponding column, based on the type of data. Besides, the unreasonable data and outliers were switched to the value of their nearest neighbor, to avoid any

potential inconsequence. We recognized the importance of this step in ensuring the reliability of our subsequent analyses and model predictions.

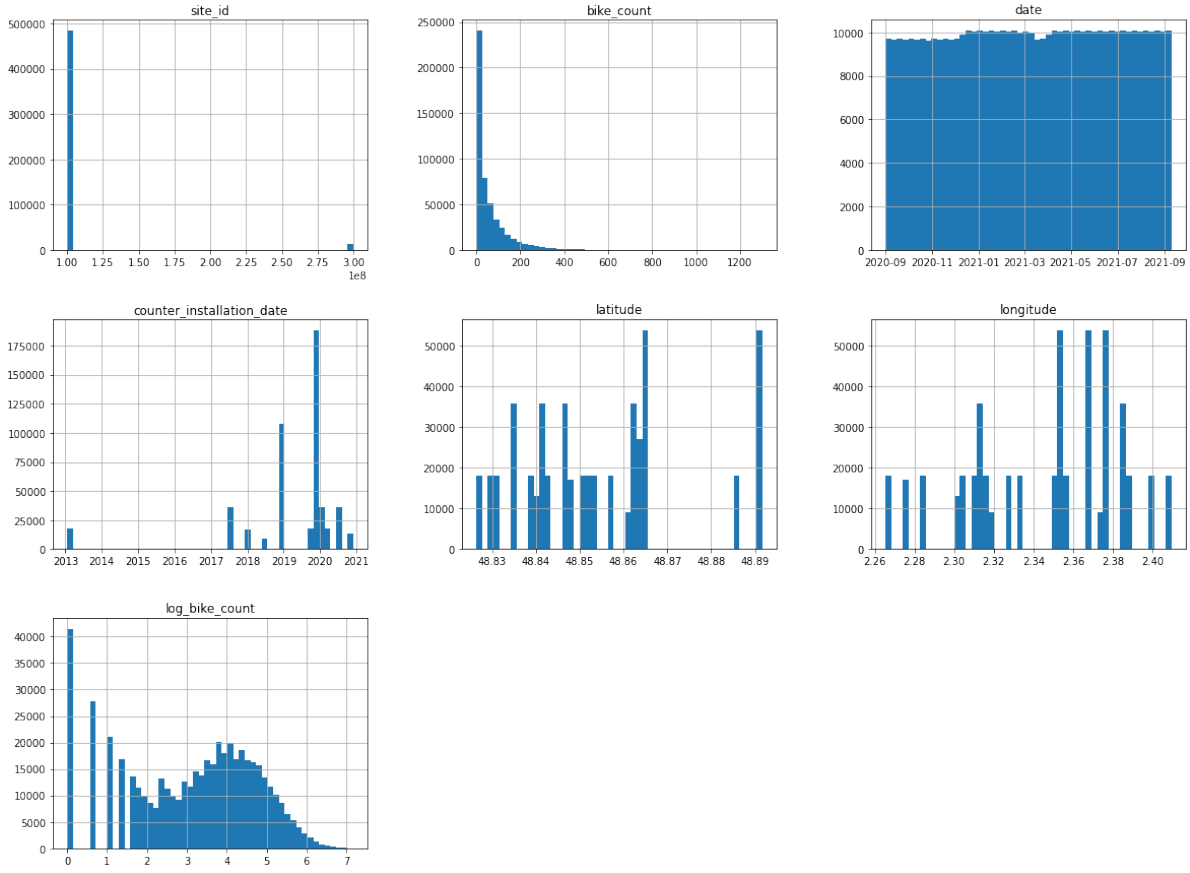

FIG. 1. Univariate Analysis was used to examine the distribution of each single variable.

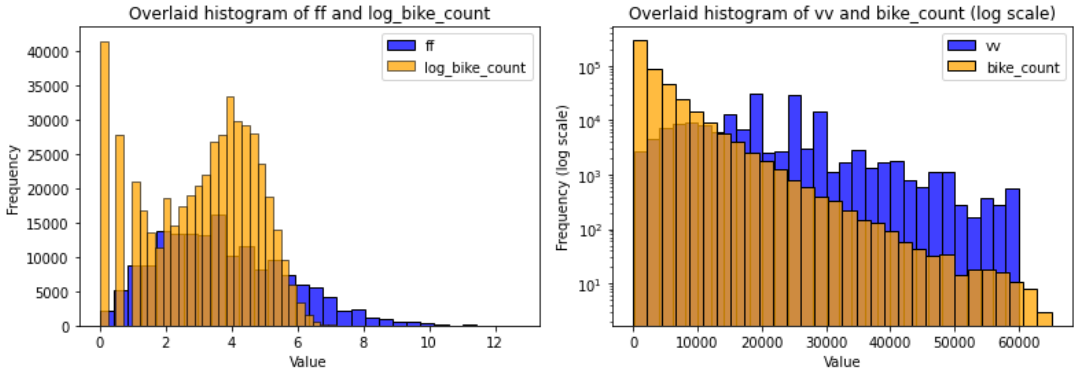

FIG. 2. Bivariate Analysis was used to explores the potential influence factors of cyclist traffic. In this plot vv represents the horizental visibility and ff represents the wind speed.

II. Exploratory Data Analysis

Our EDA includes uni-variate analysis (as shown in FIG. 1) and bi-variate analysis (as shown in FIG. 2). Using libraries such as Matplotlib and Seaborn, we conducted an extensive exploratory data analysis to understand the underlying patterns and trends

in the data. This involved visualizing various aspects such as temporal trends, spatial distribution, and the influence of external factors like weather on cyclist traffic. Also, we commented all these analysis codes in the final version to avoid unnecessary calculations.

III. Feature Engineering

Based on insights from EDA, we performed tailored feature engineering to capture the nuances of cyclist behavior in an urban setting and enhance the predictive power of our model. We created new features based on the existing data, such as time-based features (e.g., hour of the day, is-weekend judgment), spatial features (distance to the city center) and weather-related features. Particularly, we binned the weather data such as temperature to avoid the curse of dimensionality [1] and heavy calculation. These new features were then processed using one-hot encoding, binarization and categorization to enable the machine learning method to work on them. To further refine our model, we included a 'counter_age' feature, indicating the time elapsed since each counter's installation. Additionally, we meticulously processed weather data, incorporating temperature, precipitation, wind speed, visibility, and cloudiness through categorization and binning. This comprehensive strategy, consistently applied to both training and test datasets, was pivotal in effectively capturing the complex patterns of cyclist traffic in Paris, ensuring our models were robust and reflective of real-world conditions.

IV. Model Development and Validation

In our quest to develop an effective predictive model for cyclist traffic in Paris, we explored various machine learning algorithms, starting with Linear Regression to establish a baseline. Linear Regression, fundamental for its simplicity, was suitable for capturing direct relationships but was limited in handling complex patterns. We then progressed to Random Forest, an ensemble method that excels in modeling non-linear relationships and reduces the risk of overfitting by averaging multiple decision trees. This approach also provided insights into the most influential features. We also experimented with K-Nearest Neighbors (KNN) but ultimately focused on fine-tuning XGBoost using Grid Search for hyperparameter optimization. XGBoost stood out for its efficiency, scalability, and ability to handle sparse data. Its built-in regularization helped in preventing overfitting, and its flexibility allowed for custom optimization objectives and evaluation criteria. In the final phase, we trained the XGBoost model on a substantial portion of the dataset, followed by validation on a separate test set to ensure reliability and generalizability. The model's performance was evaluated using the root mean squared error (RMSE) metric, which quantifies the difference between predicted and actual values. This process confirmed XGBoost's suitability for our study, making it the ideal choice due to its accuracy and robustness in analyzing and predicting urban cycling patterns.

V. Challenges, Learnings and Improvements

In our project, we faced a major challenge in integrating external weather data, which, despite considerable efforts, did not yield the anticipated improvements in our model's performance. This experience underscored the critical role of feature engineering, revealing its substantial impact on model outcomes. Through this process, we gained valuable practical experience in using the XGBoost model and learned the importance of teamwork and collaboration in data science. Reflecting on potential improvements, we identified that exploring additional, particularly weather-related datasets for Paris and implementing normalization techniques in feature engineering could further enhance our model's accuracy. Our methodology of starting with simpler models and progressively moving to more complex ones was notably effective, allowing us to incrementally understand and improve our data predictions, thus refining our overall approach.

CONCLUSION

In concluding this report on our Kaggle Challenge project to predict cyclist traffic in Paris, we reflect on a journey marked by strategic model selection, rigorous data preprocessing, and insightful learnings. Starting with a simple Linear Regression model, progressing through Random Forest, and culminating in the implementation of XGBoost, our approach was methodical and adaptive. This progression not only enhanced our understanding of the dataset but also allowed us to fine-tune our predictions with increasing complexity and sophistication. The project underscored the pivotal role of feature engineering in predictive modeling. By carefully crafting features like time-based and geospatial attributes, and attempting to integrate external weather data, we gained a deeper appreciation of how nuanced data transformations can significantly impact model outcomes. Although

integrating external weather data posed a challenge, it offered valuable insights into dealing with diverse data sources. Our experience with XGBoost was particularly enlightening. The model's robustness, efficiency, and ability to handle various data types efficiently led to the most accurate predictions, as evidenced by the lowest RMSE scores. The grid search method for hyperparameter tuning further refined our model, demonstrating the importance of meticulous parameter optimization. In future endeavors, we recognize the potential for exploring additional datasets and employing normalization techniques in feature engineering to further enhance model performance. The learnings from this project extend beyond technical skills, encompassing teamwork, problem-solving, and balancing simplicity with complexity in model selection. This project was not just an academic exercise but a practical foray into the world of data science, offering a real-world application of Python programming, data analysis, and machine learning techniques. It has been a valuable step in our ongoing journey as aspiring data scientists, equipping us with the skills and confidence to tackle similar challenges in the future.

REFERENCES

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics).* 1st ed. Springer, 2007. ISBN: 0387310738.