Forecasting Urban Bike-Sharing Demand Using Weather and Calendar Data

A Comparison of Count Regression and Tree Ensembles

Master Thesis

Submitted as part of the requirements for the degree of Master of Science in Data Science

Author	Laura Bennett
Supervisor	Prof. Dr. Martin Keller
Date of Submission	30 September 2026

Abstract

Bike-sharing systems only work well when bikes are available where and when people want them. Operators therefore need reliable short-term forecasts of demand. This thesis compares four approaches for forecasting the hourly number of rentals in the bike-sharing system of the city of Lakeport: a seasonal naive benchmark, a Poisson generalised linear model, a random forest and gradient boosted regression trees. The models use only calendar variables and weather data, which in operation would be available a day in advance as weather forecasts. They are trained on 18 months of hourly data from 2024 and 2025 and evaluated on the following six months, which none of them has seen during training.

All three statistical models clearly outperform the benchmark. Gradient boosting achieves the lowest error, with a root mean squared error of 22.1 rentals per hour. This is 57 per cent below the benchmark and 15 per cent below the Poisson model. The advantage of the tree ensembles comes mainly from interactions between time of day, type of day and temperature, which a linear model without interaction terms cannot represent. In rainy weekend hours, however, the simpler Poisson model remains more accurate. The thesis concludes that gradient boosting is a suitable default for operational forecasts, provided that it is monitored in rare weather situations.

Keywords: bike-sharing, demand forecasting, count data, gradient boosting, random forest, weather

Acknowledgements

About this document

This is a fictional sample thesis, written to show what Obsitex makes of an Obsidian vault. The city of Lakeport, its bike-sharing system, the people named and all data are invented. The data were simulated, and every number, table and figure was computed from that simulation. The cited literature is real.

I would like to thank my supervisor, Prof. Dr. Martin Keller, for his guidance, his patience and his many questions, which made this thesis better. I am grateful to the operations team of Lakeport Bike for providing the trip data and for explaining how the system works on the street. My thanks also go to the Lakeport Meteorological Office for access to their historical weather observations.

Finally, I thank my family and friends for their support during the last two years of my studies.

Contents

Abstract ii

Acknowledgements iii

List of Figures vi

List of Tables vii

List of Abbreviations viii

1. Introduction 1

1.1. Motivation 1

1.2. Problem Statement 2

1.3. Research Questions 3

2. Background and Related Work 4

2.1. Theoretical Background 4

2.1.1. Count Regression 4

2.1.2. Tree Ensembles 5

2.2. Related Work 6

3. Methodology 8

3.1. Research Design 8

3.2. Data Collection 9

3.2.1. Trip Data 9

3.2.2. Weather Data 10

3.3. Data Analysis 10

3.3.1. Models 10

3.3.2. Evaluation 11

4. Results 13

4.1. Descriptive Results 13

4.2. Main Findings 15

5. Discussion **18**
5.1. Interpretation 18
5.2. Limitations 19

6. Conclusion and Future Work **21**
6.1. Summary of Contributions 21
6.2. Future Work 21

Bibliography **23**

Appendix **25**

A. Variable Dictionary **26**

B. Model Settings **29**

Declaration of Authorship **30**

List of Figures

3.1. Research design of the study 8

4.1. Mean hourly rentals on workdays and weekends 14
4.2. Daily rentals and daily mean temperature 14
4.3. Forecast error of the four models 16
4.4. Observed and predicted rentals in one test week 16
4.5. Permutation importance of the features 17

List of Tables

4.1. Descriptive statistics of the hourly data 13
4.2. Rentals by weather situation 15
4.3. Forecast accuracy of the models on the test period 15
4.4. Forecast error by type of day and weather 17

A.1. Variables of the hourly data set 26

B.1. Hyperparameters of the trained models 29

List of Abbreviations

- API – Application Programming Interface
- CV – Cross-Validation
- GBM – Gradient Boosting Machine
- GLM – Generalised Linear Model
- MAE – Mean Absolute Error
- NB – Negative Binomial
- RF – Random Forest
- RMSE – Root Mean Squared Error
- UTC – Coordinated Universal Time

1. Introduction

Cities around the world have introduced public bike-sharing systems to reduce car traffic, to close the last mile between home and public transport, and to offer a cheap and healthy way of getting around [5, 15]. The success of such a system depends on a simple promise: a bike is available when a customer arrives at a station, and a free dock is available at the destination. Keeping this promise requires the operator to anticipate demand. Forecasts of the number of rentals in the next hours and days drive the redistribution of bikes by van, the planning of maintenance, and the staffing of the service team.

This thesis develops and compares forecasting models for the hourly demand of the bike-sharing system in Lakeport, a mid-sized city with about 48 stations. The chapter first explains why such forecasts matter (section 1.1), then states the problem in more detail (section 1.2 on the next page) and finally formulates the research questions that guide the thesis (section 1.3 on page 3).

The remainder of the thesis is structured as follows. The statistical foundations and earlier studies are introduced in chapter 2 on page 4. The data, the models and the evaluation protocol are described in chapter 3 on page 8. The descriptive analysis and the forecast accuracy of the models follow in chapter 4 on page 13, and they are interpreted and put into perspective in chapter 5 on page 18. Finally, the contributions are summarised and directions for further research are outlined in chapter 6 on page 21.

1.1. Motivation

Demand for shared bikes is highly variable. On a warm workday, thousands of people use the system on their way to work and back home. On a cold and rainy Sunday, the same stations may stay almost untouched. Within a single day, the demand at a station can change direction several times: in the morning, bikes flow from residential areas to the business district, in the evening they flow back.

For the operator, this variability has a direct cost. Empty stations frustrate customers and lead to lost rentals, while full stations prevent customers from returning their bikes. Both problems are addressed by rebalancing, that is, by moving bikes between stations with service vans. Rebalancing is expensive: it requires vehicles, drivers and time, and it is most needed exactly during the peak hours when traffic is heaviest. Good forecasts allow the operator to plan these trips in advance rather than react to empty stations.

Lakeport Bike, the operator of the system studied in this thesis, currently plans rebalancing with a simple rule of thumb: it assumes that demand in a given hour will be the same as in the same hour one week earlier. This rule ignores the weather, which every cyclist knows to be decisive, and it fails after public holidays and during sudden changes in temperature. The operator therefore asked whether a statistical model could provide better forecasts with information that is already available the day before.[1]

1.2. Problem Statement

Forecasting bike-sharing demand is a regression problem with three properties that make it more difficult than it first appears.

First, the target is a count. The number of rentals in an hour is a non-negative integer, and its variance grows with its mean. Standard linear regression assumes a constant variance and can predict negative values, so it is not well suited for this kind of data [3].

Second, the effects of the explanatory variables are not additive. Warm weather increases demand much more on weekends, when people ride for leisure, than on workdays, when most trips are commutes that happen regardless of the temperature. Rain reduces demand most strongly during the rush hours. Such interactions have to be specified by hand in a classical regression model, whereas tree-based methods can learn them from the data [9].

Third, the data form a time series. A model that is evaluated on randomly selected hours will see the future during training, which leads to overly optimistic error estimates [1]. A fair comparison therefore has to respect the order of time.

[1] Weather forecasts for the next 24 hours are available to the operator every evening through the API of the Lakeport Meteorological Office.

The problem addressed in this thesis is to find out which type of model forecasts the hourly demand of the Lakeport system most accurately under these conditions, and to understand where the differences between the models come from.

1.3. Research Questions

Based on the problem statement, the thesis answers the following three research questions:

1. How accurately can the hourly demand of the Lakeport bike-sharing system be forecast with calendar and weather information alone?
2. Do tree ensembles (random forest and gradient boosting) forecast more accurately than a Poisson generalised linear model, and by how much?
3. Which variables contribute most to the forecasts, and in which situations does each model perform best or worst?

To answer these questions, the thesis makes four contributions:

- a cleaned hourly data set that combines two years of trip records with weather observations,
- a forecasting pipeline that respects the time order of the data in both training and evaluation,
- a systematic comparison of four models on a hold-out period of six months,
- an analysis of forecast errors by type of day and weather, which shows where the models differ.

2. Background and Related Work

This chapter lays the foundations for the models used later. Regression models for count data and the two families of tree ensembles that are compared in this thesis are introduced in section 2.1. Research on bike-sharing systems and on forecasting their demand is then reviewed in section 2.2 on page 6.

2.1. Theoretical Background

The models compared in this thesis come from two traditions. Count regression extends the linear model to data that consist of non-negative integers (section 2.1.1). Tree ensembles combine many simple decision trees into one flexible prediction (section 2.1.2 on the next page). Both are introduced here only as far as they are needed for the following chapters. A thorough treatment can be found in the textbook by Hastie, Tibshirani and Friedman [9].

2.1.1. Count Regression

Let y_t denote the number of rentals in hour t and $\mathbf{x}_t$ the vector of explanatory variables for that hour. A natural model for count data is the Poisson distribution, whose probability mass function is

$$P(Y_t = y \mid \mathbf{x}_t) = \frac{\mu_t^y \, e^{-\mu_t}}{y!}, \qquad y = 0, 1, 2, \ldots$$

The Poisson generalised linear model links the expected value μ_t to the explanatory variables through a logarithm [13]:

$$\log \mu_t = \beta_0 + \sum_{j=1}^{p} \beta_j \, x_{tj}$$

The log link guarantees positive forecasts and turns the effect of each variable into a multiplicative factor: if x_{tj} increases by one unit, the expected demand is multiplied by e^{β_j}. The coefficients are estimated by maximising the log-likelihood, usually with a small penalty $\alpha\|\boldsymbol{\beta}\|_2^2$ that stabilises the estimate when many variables are correlated.

A limitation of the Poisson model is that it assumes the variance to be equal to the mean. Real count data are often overdispersed, which means that the variance exceeds the mean. The negative binomial distribution relaxes this assumption with an additional dispersion parameter k [3]:

$$\operatorname{Var}(Y_t) = \mu_t + \frac{\mu_t^2}{k}$$

For large k the negative binomial distribution approaches the Poisson distribution. Overdispersion does not bias the Poisson estimates of $\boldsymbol{\beta}$, but it makes their standard errors too small. Since this thesis is concerned with forecasts and not with significance tests, the Poisson model remains a reasonable choice.

2.1.2. Tree Ensembles

A regression tree partitions the space of explanatory variables into rectangular regions and predicts a constant value in each region. Single trees are easy to interpret but unstable: a small change in the data can produce a very different tree. Ensembles combine many trees to obtain more stable and more accurate predictions [9].

A **random forest** averages B trees, each grown on a bootstrap sample of the training data and with a random subset of variables considered at each split [2]:

$$\hat{f}_{\mathrm{RF}}(\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B} T_b(\mathbf{x})$$

Gradient boosting builds the ensemble sequentially [8]. Each new tree T_m is fitted to the negative gradient of a loss function L at the current prediction, and is added with a small learning rate ν:

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \nu\, T_m(\mathbf{x}), \qquad T_m \approx -\left.\frac{\partial L\big(y, F(\mathbf{x})\big)}{\partial F(\mathbf{x})}\right|_{F=F_{m-1}}$$

For count data, the Poisson deviance is a suitable loss function. The model then predicts $\log \mu_t$, exactly like the Poisson generalised linear model, but replaces the linear predictor by a sum of trees. Because each tree can split on several variables in a row, interactions between variables are learned automatically.

Modern implementations such as XGBoost [4] and LightGBM [12] make gradient boosting fast enough for data sets with millions of rows. They bin each variable into a small number of intervals before searching for splits, which reduces the cost of training considerably. This thesis uses the histogram-based gradient boosting of the Python library scikit-learn [14], which follows the same idea.

2.2. Related Work

Public bike-sharing has a history of more than half a century. DeMaio traces it from the first free bicycles in Amsterdam in the 1960s to the coin-deposit systems of the 1990s and the IT-based systems with smart cards and docking stations that spread in the 2000s [5]. Shaheen, Guzman and Zhang describe the same development as a sequence of generations and show how the systems spread across Europe, the Americas and Asia [15]. In a later review, Fishman summarises what is known about who uses bike-sharing, for which purposes, and what barriers keep people from using it [7]. Weather and the time of day are among the factors that this literature repeatedly links to usage.

The data side of the field was shaped to a large extent by Fanaee-T and Gama, who combined two years of hourly rental counts of the Capital Bikeshare system in Washington, D.C., with weather and holiday information [6]. Their data set became a widely used benchmark for regression methods. Its structure is very similar to the data used in this thesis: an hourly count as target, and calendar and weather variables as predictors.

On the methodological side, two lines of work are relevant. The first concerns models for count data, where generalised linear models with a Poisson or negative binomial distribution are the established standard [3, 13]. The second concerns tree ensembles. Random forests [2] and gradient boosting [8] have become standard tools for tabular data, and efficient implementations [4, 12] have made them practical for large data sets.

Finally, the evaluation of forecasting models raises its own questions. Hyndman and Koehler discuss which error measures are appropriate for which kind of series [11]. Bergmeir and Benítez study how cross-validation can be applied to time series

without letting information from the future leak into the training data [1]. The textbook by Hyndman and Athanasopoulos gives a practical introduction to both topics [10].

Compared with this literature, the present thesis does not propose a new method. Its contribution is a careful comparison of established methods on a new data set, with an evaluation protocol that respects the order of time and an analysis of where the models differ.

3. Methodology

This chapter describes how the research questions are answered. An overview of the procedure is given in section 3.1. The two data sources and the way they were cleaned and combined are described in section 3.2 on the next page. The four forecasting models and the protocol used to evaluate them are presented in section 3.3 on page 10.

3.1. Research Design

The thesis follows a quantitative, comparative design. Four forecasting models are trained on the same data and evaluated on the same period, so that differences in accuracy can be attributed to the models and not to the data. The procedure consists of five steps, which are shown in figure 3.1.

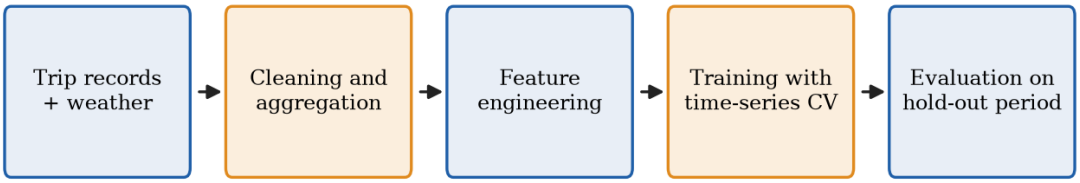


Figure 3.1.: Research design of the study

1. Trip records from the operator and hourly weather observations are collected for the years 2024 and 2025.
2. The trip records are cleaned and aggregated to the number of rentals per hour, and joined with the weather data.
3. Calendar variables are derived from the timestamp, and the explanatory variables for each model are prepared.
4. The models are trained on the first 18 months. A time-series cross-validation within this period checks whether their ranking is stable.

5. The trained models forecast the last six months, and their errors are compared.

All steps were implemented in Python with the libraries listed in appendix B on page 29. The code is kept in the folder *Data* of the project, outside the text of this thesis. It uses a fixed random seed, so that every number reported here can be reproduced.

3.2. Data Collection

The analysis combines two sources: the trip records of the bike-sharing system (section 3.2.1) and hourly weather observations (section 3.2.2 on the following page). Both cover the period from 1 January 2024 to 31 December 2025. After cleaning and aggregation, they form one table with one row per hour, 17,544 rows in total. Every variable of this table is documented in appendix A on page 26.

3.2.1. Trip Data

Lakeport Bike operates 48 docking stations in the city centre and the surrounding districts. Each rental is recorded by the system with the following information:

- a unique trip identifier,
- the start and end time, stored in UTC,[1]
- the start and end station,
- the type of bike (conventional or electric),
- the type of customer (subscriber or casual user).

Before the analysis, the trip records were cleaned with three rules. Trips shorter than one minute that started and ended at the same station were removed, because they usually indicate that a customer returned a bike immediately after noticing a defect. Trips longer than 24 hours were removed as well, since they mostly reflect bikes that were not returned properly. Finally, trips made by service staff for rebalancing or maintenance were excluded, because they do not represent customer demand.

[1] All timestamps were converted to local time before the hourly aggregation, so that the hour of the day matches the daily routine of the customers, also across the changes to and from daylight saving time.

The remaining trips were counted per hour of their start time. The resulting series contains 1,042,769 rentals over two years, or on average 1,426 rentals per day. The busiest day saw 3,984 rentals.

Personal data were not part of the data set. The operator removed all customer identifiers before handing over the records.[2]

3.2.2. Weather Data

The Lakeport Meteorological Office provides hourly observations from its station near the city centre through a public API. For each hour, the following variables were retrieved: air temperature in degrees Celsius, relative humidity in per cent, mean wind speed in kilometres per hour, and precipitation in millimetres.

From these observations, a categorical variable *weather situation* was derived with four levels. An hour counts as *heavy rain* if more than 2.5 mm of precipitation fell, and as *light rain* if any smaller amount fell. Dry hours are classified as *cloudy or misty* or as *clear* according to the cloud cover reported by the station for that day.

In practice, an operator would plan with weather forecasts, not with observations. Historical forecasts were not available for the study period, so the observations serve as a stand-in. The forecast errors reported in this thesis are therefore somewhat lower than they would be in operation. This point is discussed further in section 5.2 on page 19.

3.3. Data Analysis

The analysis compares four models of increasing flexibility. Each model and the variables it receives are described in section 3.3.1. How the data are split in time, how the models are cross-validated, and which measures are used to compare their forecasts is explained in section 3.3.2 on the following page.

3.3.1. Models

Seasonal naive benchmark. The first model reproduces the current practice of the operator. It forecasts the demand of an hour with the observed demand of the same hour one week earlier:

[2]The data were provided under a data use agreement that permits their use for this thesis only.

$$\hat{y}_t = y_{t-168}$$

This benchmark needs no training and captures both the daily and the weekly cycle. Every other model has to beat it to be useful.

Poisson generalised linear model. The second model is the Poisson model with log link described in section 2.1.1 on page 4. Since a linear predictor cannot learn interactions on its own, the hour of the day enters the model separately for workdays and for weekends and holidays, as 46 indicator variables. The annual cycle is represented by a sine and a cosine of the day of the year. Further variables are the weather situation (as indicators), temperature and squared temperature, humidity, wind speed, precipitation, a holiday indicator and the year. All variables are standardised with the mean and standard deviation of the training data, and a small ridge penalty with $\alpha = 10^{-4}$ is applied.

Random forest. The third model is a random forest with 300 trees [2]. It receives the twelve raw variables listed below, without any indicator coding or interaction terms, since trees can split on any variable directly:

- calendar: hour, weekday, workday, holiday, month, season, year,
- weather: temperature, humidity, wind speed, precipitation, weather situation.

Gradient boosting. The fourth model uses gradient boosted trees with the Poisson deviance as loss function, as described in section 2.1.2 on page 5. It receives the same twelve variables as the random forest.

The hyperparameters of the three trained models were fixed in advance to common values and are listed in appendix B on page 29. They were not tuned on the test period. The cross-validation described in section 3.3.2 was used only to check that the ranking of the models does not depend on a particular part of the training period.

3.3.2. Evaluation

Split in time. The data are split into a training period from 1 January 2024 to 30 June 2025 (13,128 hours) and a test period from 1 July to 31 December 2025 (4,416 hours). All models are trained on the training period only and then forecast the entire test period. This mirrors the situation of the operator, who has to forecast hours that lie in the future.

Time-series cross-validation. Within the training period, the stability of the results is checked with an expanding window [1]. Four validation windows of 91 days each start on 1 July 2024, 1 October 2024, 1 January 2025 and 1 April 2025. For each window, the models are trained on all hours before its start and evaluated on the hours inside it. In contrast to ordinary cross-validation, no model ever sees data from after the hours it is evaluated on.

Error measures. Let n be the number of hours in the evaluation period, y_t the observed and $\hat{y}_t$ the forecast number of rentals. Three measures are reported [11]. The root mean squared error penalises large errors strongly and is the main criterion of this thesis:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}$$

The mean absolute error is less sensitive to single large errors and can be read directly as the average number of rentals by which a forecast misses:

$$\mathrm{MAE} = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t|$$

The coefficient of determination relates the squared error to the variance of the observations, where $\bar{y}$ is the mean of the observed values:

$$R^2 = 1 - \frac{\sum_t (y_t - \hat{y}_t)^2}{\sum_t (y_t - \bar{y})^2}$$

Percentage errors such as the mean absolute percentage error are not used, because they are undefined for hours without any rental, and such hours occur every night.

Variable importance. To understand which variables drive the forecasts of the best model, permutation importance is computed on the test period [2]. The values of one variable are shuffled randomly, which breaks its relation to the target, and the resulting increase in RMSE is recorded. The procedure is repeated five times per variable and the increases are averaged.

4. Results

This chapter presents the results in two steps. The data and the patterns of demand that any good model has to reproduce are described in section 4.1. The forecast accuracy of the four models is then compared in section 4.2 on page 15, together with an analysis of where they differ.

4.1. Descriptive Results

Over the two years, the system recorded 1,042,769 rentals. The target and the four continuous weather variables are summarised in table 4.1. On average, 59.4 bikes were rented per hour, but the standard deviation of 85.4 shows how much this number varies. The variance of the hourly counts is about 123 times larger than their mean, a clear sign of overdispersion. In 3.6 per cent of all hours, mostly at night, not a single bike was rented.

Variable	Mean	Std. dev.	Minimum	Maximum
Hourly rentals	59.4	85.4	0	767
Temperature (°C)	11.3	8.1	−8.0	30.8
Relative humidity (%)	69.3	10.9	28.4	100.0
Wind speed (km/h)	12.0	7.8	0.1	70.0
Precipitation (mm/h)	0.17	0.61	0.00	9.07

Table 4.1.: Descriptive statistics of the hourly data

The strongest pattern in the data is the daily cycle, which is shown in figure 4.1 on the following page. On workdays, demand has two sharp peaks: one at 8:00 in the morning and a higher one at 17:00 in the afternoon. These are the commuting trips. On weekends and holidays, the curve has a completely different shape, with a single broad peak around 14:00. Any model that treats all days alike will therefore make large errors.

The second strongest influence is the temperature. The number of rentals per day is plotted against the daily mean temperature in figure 4.2 on the next page. Demand

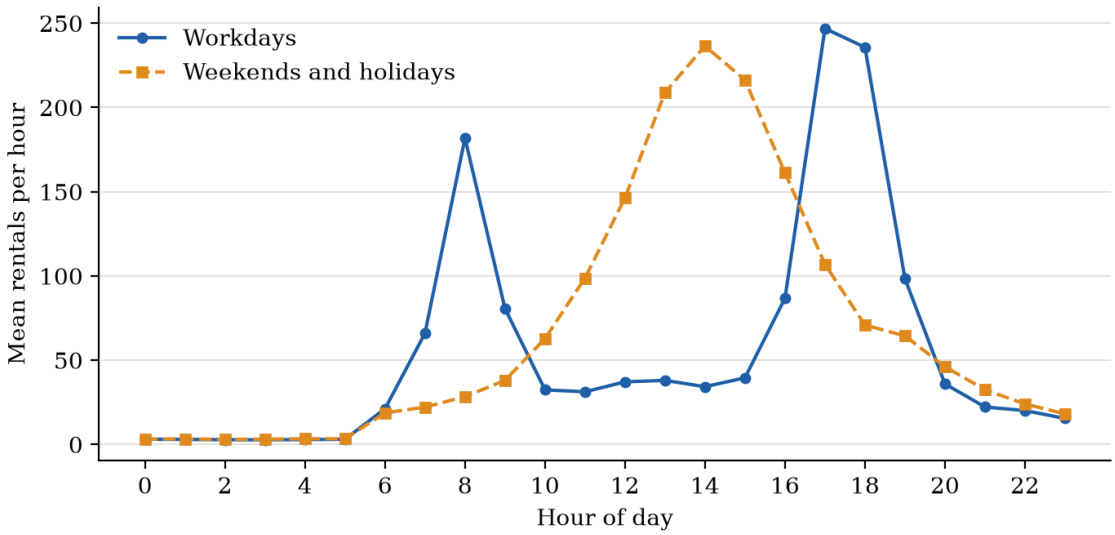


Figure 4.1.: Mean hourly rentals on workdays and weekends

rises steadily from about 500 rentals on days around freezing point to well over 2,000 rentals on warm days. Above roughly 23 °C the increase levels off. The scatter around the mean is much larger on warm days than on cold days, which is consistent with the overdispersion noted above. Part of this scatter is explained by the type of day: warm weekends attract many more leisure riders than warm workdays.

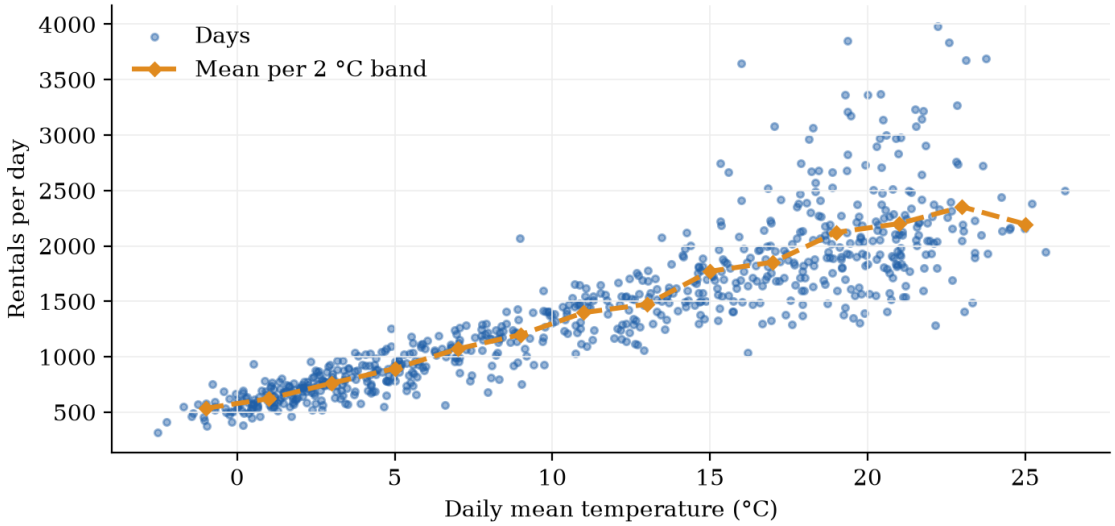


Figure 4.2.: Daily rentals and daily mean temperature

Precipitation has a clear negative effect, as table 4.2 on the following page shows. In hours with light rain, the mean demand is 44 per cent below that of clear hours. In hours with heavy rain, it drops by more than three quarters. Heavy rain is rare, however, and occurred in only 1.7 per cent of all hours.

Weather situation	Share of hours (%)	Mean rentals	Median rentals
Clear	62.1	66.6	31
Cloudy or misty	25.5	54.4	26
Light rain	10.8	37.1	19
Heavy rain	1.7	15.9	9

Table 4.2.: Rentals by weather situation

Finally, demand grew over time. In 2025, the system recorded 10.5 per cent more rentals than in 2024. All models therefore receive the year as an explanatory variable.

4.2. Main Findings

Overall accuracy. The three error measures for all models on the six-month test period are reported in table 4.3, together with the mean RMSE of the time-series cross-validation. All three trained models are far more accurate than the seasonal naive benchmark. Gradient boosting achieves the best value on every measure. Its RMSE of 22.1 rentals per hour is 57 per cent lower than that of the benchmark and 15 per cent lower than that of the Poisson model.

Model	RMSE	MAE	R^2	CV RMSE
Seasonal naive	51.7	26.0	0.691	–
Poisson GLM	25.8	13.4	0.923	22.2
Random forest	23.4	12.4	0.937	25.8
Gradient boosting	**22.1**	**11.6**	**0.944**	**21.3**

Table 4.3.: Forecast accuracy of the models on the test period

The same ranking can be seen at a glance in figure 4.3 on the following page. The cross-validation confirms the first place of gradient boosting, which had the lowest error in three of the four validation windows. The order of the Poisson model and the random forest, however, is not stable: the random forest is better on the test period but worse in the cross-validation, where it had the highest error of the three models in three of the four windows.

Forecasts over time. The forecasts of gradient boosting and of the Poisson model are compared with the observed demand in the week from 8 to 14 September 2025 in figure 4.4 on the next page. Both models reproduce the double peak of the workdays and the single peak of the weekend well. The Poisson model tends to overshoot the evening peaks on warm days, while gradient boosting follows them more closely.

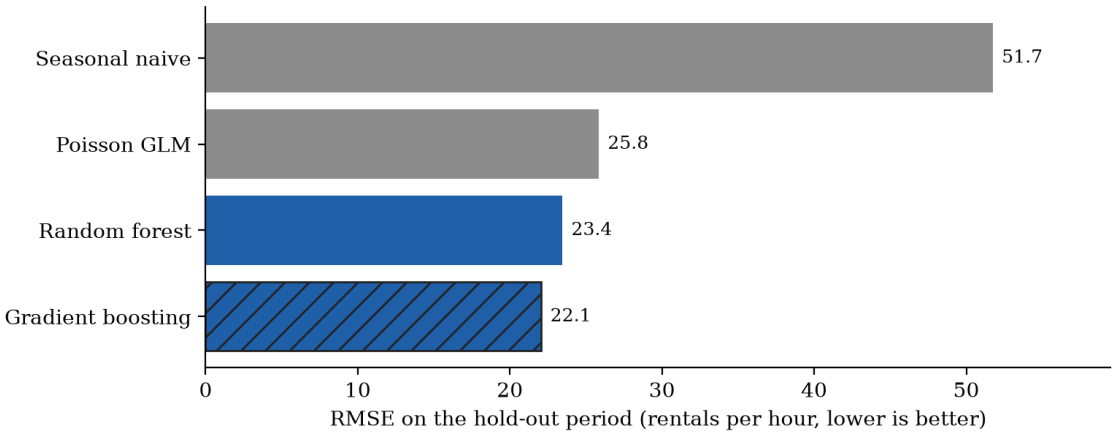


Figure 4.3.: Forecast error of the four models

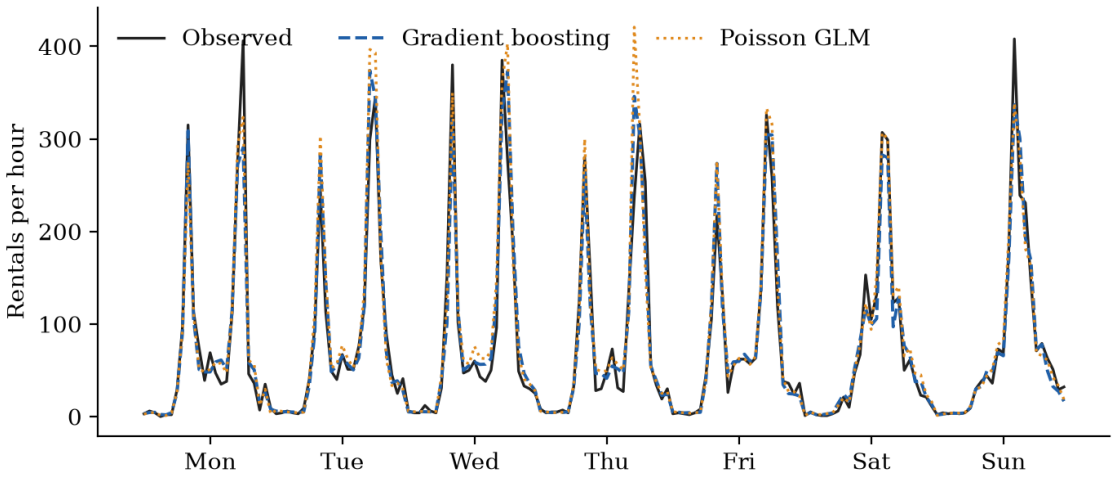


Figure 4.4.: Observed and predicted rentals in one test week

Errors by type of hour. To find out where the advantage of the tree ensembles comes from, the test period was divided by type of day and by weather, where weekends include public holidays. The RMSE of the three trained models in each group is shown in table 4.4. Gradient boosting is the most accurate model in three of the four groups, with the largest gain over the Poisson model on dry workdays. In the 227 rainy weekend hours, however, the Poisson model is clearly better than both tree ensembles.

Type of hour	Hours	GLM	RF	GBM
Workdays, dry	2,712	25.8	21.2	**20.7**
Workdays, wet	384	17.5	19.5	**15.5**
Weekends, dry	1,093	29.7	29.0	**27.3**
Weekends, wet	227	**15.2**	24.7	18.5

Table 4.4.: Forecast error by type of day and weather

Important variables. The permutation importance of the twelve variables for gradient boosting is shown in figure 4.5. The hour of the day is by far the most important variable, followed by the type of day and the temperature. Shuffling the hour raises the RMSE by about 90 rentals, shuffling the workday indicator by about 49 and the temperature by about 38. Precipitation and wind speed follow at a distance. Month and season contribute almost nothing, because the temperature already carries most of the seasonal information. The year has no importance at all on the test period, simply because it is constant there.

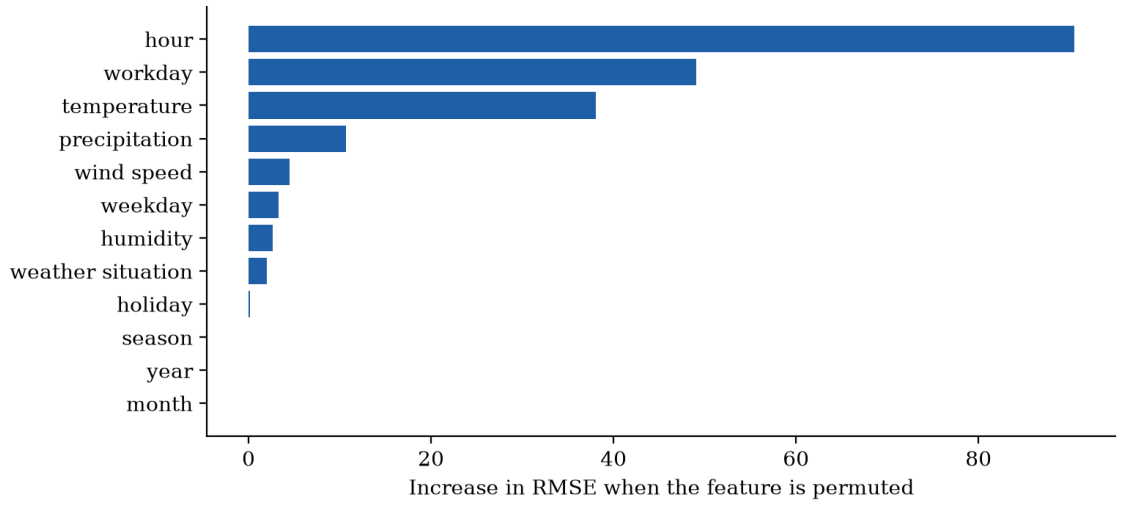


Figure 4.5.: Permutation importance of the features

5. Discussion

This chapter interprets the results with respect to the three research questions and discusses what they mean for the operator (section 5.1). It then names the limitations of the study that a reader should keep in mind (section 5.2 on the next page).

5.1. Interpretation

Research question 1 asked how accurately the hourly demand can be forecast with calendar and weather information alone. The answer is: quite accurately. The best model explains 94 per cent of the variance in the test period, and its forecasts miss the observed demand by 11.6 rentals per hour on average, compared with a mean demand of 59.4. For the planning of rebalancing, which happens in blocks of several hours, errors of this size are small enough to be useful.

Research question 2 asked whether tree ensembles outperform the Poisson model. For gradient boosting, the answer is a clear yes: it reduces the RMSE by 15 per cent and was the best model in the cross-validation as well. For the random forest, the answer is less clear. It was 9 per cent more accurate than the Poisson model on the test period, but less accurate in the cross-validation. One plausible reason is that the random forest predicts averages of observed counts and has no notion of a multiplicative effect, so it adapts less well when the level of demand shifts between periods.

Research question 3 asked which variables matter most and where the models differ. The time of day, the type of day and the temperature dominate, which agrees with the descriptive analysis in section 4.1 on page 13. The analysis by type of hour shows that the advantage of gradient boosting is largest on dry workdays and dry weekends. These are exactly the situations in which the effect of the temperature depends on the hour and on the type of day. A warm evening adds many leisure trips, a warm morning adds few. Gradient boosting learns such interactions from the data, while the Poisson model would need them to be specified by hand.

The weakness of the tree ensembles in rainy weekend hours has a simple explanation. Only about five per cent of all hours fall into this group, so the trees have few examples to learn from, and each leaf averages over a small and noisy sample. The Poisson model, in contrast, estimates one effect of rain for all hours and multiplies it with the level of demand. This structural assumption is not perfectly true, but it generalises better when data are scarce.

For the operator, the results suggest a pragmatic approach. In an interview conducted for this thesis, the head of operations of Lakeport Bike described the current situation as follows:

> We know that the weekly rule fails whenever the weather changes. On those days, our drivers spend the morning chasing empty stations instead of preparing for the evening peak.

Gradient boosting would replace the weekly rule as the default forecast. Its errors in rare weather situations should be monitored, and in such hours the Poisson model could serve as a second opinion.

5.2. Limitations

The results of this thesis are subject to several limitations:

- **Observed instead of forecast weather.** The models use the weather that actually occurred. In operation, only weather forecasts are available, and their errors add to the errors of the demand model. The accuracy reported here is therefore an upper bound.
- **One city, two years.** The data come from a single system over two years. Patterns such as the strong commuting peaks may be weaker in cities with more tourism, and the models may need to be retrained after changes in the network.
- **System level instead of station level.** The thesis forecasts the total demand of the system. Rebalancing decisions are made per station, where counts are smaller and noisier, and where the direction of the flows matters as much as their size.
- **No tuning of hyperparameters.** The models were run with fixed settings. A systematic search might improve all three trained models, and it might change the order of the Poisson model and the random forest.

- **No special events.** Concerts, strikes in public transport or road closures can change demand strongly for a few hours. Such events are not part of the data.
- **Point forecasts only.** The models predict one number per hour. For planning, a forecast interval would often be more useful than a single value.

Despite these limitations, the comparison is fair in one important respect: all models were trained and evaluated on exactly the same data and under the same rules, so the differences between them are not an artefact of the setup.

6. Conclusion and Future Work

This thesis set out to find the most accurate way of forecasting the hourly demand of the Lakeport bike-sharing system with calendar and weather information. Four models were compared on a test period of six months that none of them had seen during training. Gradient boosted regression trees with a Poisson loss proved to be the most accurate model, both on the test period and in a time-series cross-validation.

6.1. Summary of Contributions

The thesis makes the following contributions:

- It combines two years of trip records and weather observations into one hourly data set and documents every variable of it.
- It shows that a simple statistical model already halves the error of the weekly rule that the operator currently uses.
- It quantifies the additional gain of gradient boosting at 15 per cent of the RMSE of a Poisson model, and traces this gain back to interactions between time and temperature.
- It identifies rainy weekend hours as a weak spot of the tree ensembles, where the more structured Poisson model is more reliable.

6.2. Future Work

Several extensions follow directly from the limitations discussed in section 5.2 on page 19. The most important is to replace the observed weather by archived weather forecasts, which would show how much accuracy is lost in operation. A second step is to move from the system level to the level of single stations, possibly with a hierarchical model that shares information between neighbouring stations. Third,

the models could be extended to produce forecast intervals, for example with a quantile loss, so that the operator can plan for a range of outcomes rather than a single number. Finally, a combination of gradient boosting and the Poisson model, weighted by the weather situation, could unite the strengths of both.

Bibliography

[1] Christoph Bergmeir and José M. Benítez. "On the Use of Cross-Validation for Time Series Predictor Evaluation". In: *Information Sciences* 191 (2012), pp. 192–213.

[2] Leo Breiman. "Random Forests". In: *Machine Learning* 45.1 (2001), pp. 5–32.

[3] A. Colin Cameron and Pravin K. Trivedi. *Regression Analysis of Count Data*. 2nd ed. Cambridge: Cambridge University Press, 2013.

[4] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, pp. 785–794.

[5] Paul DeMaio. "Bike-sharing: History, Impacts, Models of Provision, and Future". In: *Journal of Public Transportation* 12.4 (2009), pp. 41–56.

[6] Hadi Fanaee-T and João Gama. "Event Labeling Combining Ensemble Detectors and Background Knowledge". In: *Progress in Artificial Intelligence* 2.2 (2014), pp. 113–127.

[7] Elliot Fishman. "Bikeshare: A Review of Recent Literature". In: *Transport Reviews* 36.1 (2016), pp. 92–113.

[8] Jerome H. Friedman. "Greedy Function Approximation: A Gradient Boosting Machine". In: *The Annals of Statistics* 29.5 (2001), pp. 1189–1232.

[9] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. New York: Springer, 2009.

[10] Rob J. Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. 3rd ed. Melbourne: OTexts, 2021.

[11] Rob J. Hyndman and Anne B. Koehler. "Another Look at Measures of Forecast Accuracy". In: *International Journal of Forecasting* 22.4 (2006), pp. 679–688.

[12] Guolin Ke et al. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In: *Advances in Neural Information Processing Systems 30*. 2017.

[13] Peter McCullagh and John A. Nelder. *Generalized Linear Models*. 2nd ed. London: Chapman and Hall, 1989.

[14] Fabian Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[15] Susan A. Shaheen, Stacey Guzman, and Hua Zhang. "Bikesharing in Europe, the Americas, and Asia: Past, Present, and Future". In: *Transportation Research Record* 2143 (2010), pp. 159–167.

Appendix

A. Variable Dictionary

The following table documents every variable of the data set, from the raw fields of the trip records and the weather data to the derived variables that enter the models. The column *Source* states where a variable comes from: the trip records of Lakeport Bike, the API of the Lakeport Meteorological Office, or a derivation during data preparation.

Table A.1.: Variables of the hourly data set

Variable	Source	Type	Description
trip_id	Trip records	Text	Unique identifier of a rental, assigned by the system.
start_time	Trip records	Timestamp	Time at which the bike was taken from the dock, in UTC.
end_time	Trip records	Timestamp	Time at which the bike was returned, in UTC.
start_station	Trip records	Category	Station at which the rental started, one of 48.
end_station	Trip records	Category	Station at which the bike was returned.
duration	Trip records	Number	Length of the rental in seconds, computed from start and end time.
bike_type	Trip records	Category	Conventional or electric bike.
customer_type	Trip records	Category	Subscriber with an annual pass, or casual user.
temperature	Weather API	Number	Air temperature in °C, mean over the hour.
humidity	Weather API	Number	Relative humidity in per cent, mean over the hour.
wind_speed	Weather API	Number	Mean wind speed in km/h.

Variable	Source	Type	Description
precipitation	Weather API	Number	Precipitation in mm, sum over the hour.
cloud_cover	Weather API	Category	Daily cloud cover class reported by the station: clear or cloudy.
weather_situation	Derived	Category	1 clear, 2 cloudy or misty, 3 light rain, 4 heavy rain (more than 2.5 mm in the hour).
timestamp	Derived	Timestamp	Start of the hour in local time. Key of the hourly table.
hour	Derived	Integer	Hour of the day, from 0 to 23.
weekday	Derived	Integer	Day of the week, from 0 (Monday) to 6 (Sunday).
holiday	Derived	Binary	1 on public holidays in Lakeport, otherwise 0.
workday	Derived	Binary	1 from Monday to Friday unless the day is a holiday, otherwise 0.
month	Derived	Integer	Month of the year, from 1 to 12.
season	Derived	Integer	Meteorological season: 1 winter, 2 spring, 3 summer, 4 autumn.
year	Derived	Integer	0 for 2024, 1 for 2025.
day_of_year	Derived	Integer	Day of the year, from 1 to 366.
doy_sin, doy_cos	Derived	Number	Sine and cosine of the day of the year. Annual cycle in the Poisson model.
temperature_sq	Derived	Number	Squared temperature. Curvature of the temperature effect in the Poisson model.
hour_workday_1 to _23	Derived	Binary	23 indicators for the hour of the day on workdays. Poisson model only.
hour_weekend_1 to _23	Derived	Binary	23 indicators for the hour of the day on weekends and holidays. Poisson model only.

Variable	Source	Type	Description
weather_2 to _4	Derived	Binary	Indicators for the weather situations 2 to 4. Poisson model only.
rentals_lag_168	Derived	Integer	Rentals in the same hour one week earlier. Forecast of the seasonal naive benchmark.
rentals	Derived	Integer	Number of rentals that started in the hour. Target variable of all models.

B. Model Settings

The settings of the three trained models are listed in table B.1. All settings were fixed before the analysis and were not tuned on the test period.

Model	Setting	Value
Poisson GLM	Ridge penalty α	10^{-4}
	Maximum iterations	1,000
	Scaling of the variables	standardised
Random forest	Number of trees	300
	Minimum hours per leaf	3
	Share of variables per split	0.6
Gradient boosting	Loss function	Poisson deviance
	Learning rate	0.06
	Boosting iterations	600
	Maximum leaves per tree	31
	Minimum hours per leaf	20
	L2 regularisation	1.0
All models	Random seed	1

Table B.1.: Hyperparameters of the trained models

The variables of the Poisson model were standardised with the mean and standard deviation of the training data only, so that no information from the test period entered the model.

The analysis was carried out with the following software:

- Python 3.14.6
- NumPy 2.5.3 for the data preparation
- scikit-learn 1.9.1 for all models and the permutation importance [14]
- Matplotlib 3.11.2 for the figures

Declaration of Authorship

I hereby declare that I have written this thesis independently, that I have not used sources or aids other than those stated, and that I have marked all passages taken from other works as such.

Lakeport, 30 September 2026 – Laura Bennett