



**Politecnico  
di Torino**

**POLITECNICO DI TORINO**

Master's Degree in Computer Engineering

July 2023

**Exploring micromobility dynamics  
through Machine Learning prediction  
algorithms: an analysis of urban  
transportation patterns**

Supervisors

Prof. Silvia CHIUSANO

Dr. Andrea AVIGNONE

Candidate

Giorgia CHIOTTI

## Abstract

The term "micromobility" has just recently entered our lexicon (approximately 2017) and it can be defined simply as mobility pertaining to short routes and distances, primarily in cities. Micro vehicles, which have a light mass and a constrained speed, are included in the idea of micromobility, both powered and unpowered, private and shared vehicles are covered in this list. The shared use of vehicles, specifically bicycles, is the main topic of this thesis. Micromobility sharing services have expanded significantly since this idea was first proposed in many parts of the world.

In this study, the usage of these services was analyzed using some Machine Learning models such as: ARIMA, Linear Regression, Lasso, Ridge, Random Forest and Gradient Boosting. The goal of this thesis is addressing the problem of predicting the availability of bikes for a station-based sharing service and the flux of bikes in a certain area. Specifically, it investigates the applicability of machine learning models to forecast the number of occupied slots of a particular sharing station of bikes in San Francisco (USA) and the number of bikes that cross the Fremont Bridge in Seattle (USA). To accomplish this, a methodology has been developed. First, the data was collected, after which it was cleaned, integrated, and subjected to preliminary analysis to determine how best to manage it. Next, the chosen machine learning models were trained using the sliding window technique, their performance was compared, and the best machine learning model was chosen. Lastly, by analyzing temporal and weather contexts, using the chosen Machine Learning model trained on past data, we obtained a good future prediction on bike occupancy.

The outcomes allow us to provide practical guidelines to setup and tune Machine Learning models on these fields. As a practical result of this work, it is possible to forecast the arrival of bikes or, more broadly, people in a certain urban area, which might help businesses or local governments to enhance the services they provide there.

**Keywords:** micromobility, data analysis, machine learning, ARIMA, Linear Regression, Lasso, Ridge, Random Forest, Gradient Boosting.

# Acknowledgements

Il mio primo ringraziamento va alla professoressa Silvia Chiusano e al mio correlatore Andrea Avignone, per avermi seguita in questo progetto di tesi ed essere stati sempre disponibili.

Ringrazio i miei genitori per avermi sempre spronata e supportata durante tutto il mio percorso di studi. In particolare vorrei ringraziare mio fratello Alessandro per le giornate di studio passate insieme e per essere sempre pronto a darmi una mano. Un ringraziamento va ai miei nonni, che hanno sempre creduto in me.

Infine, vorrei ringraziare tutti i miei amici e il mio ragazzo, che sono stati sempre al mio fianco e sono riusciti ad alleggerirmi gli anni di studio attraverso i momenti passati insieme.



# Table of Contents

<b>List of Tables</b>	v
<b>List of Figures</b>	vi
<b>Acronyms</b>	x
<b>1 Introduction</b>	1
1.1 Context . . . . .	1
1.2 Thesis outline . . . . .	2
<b>2 Urban mobility data</b>	4
2.1 Mobility and Micromobility . . . . .	4
<b>3 Data description</b>	10
3.1 Station-based SF Bay Area Bike Share . . . . .	10
3.2 Seattle: Fremont Bridge Bicycle Counter . . . . .	12
<b>4 Data analysis framework and methodology</b>	13
4.1 Framework . . . . .	13
4.2 Proposed Methodology . . . . .	14
4.2.1 Data Acquisition . . . . .	14
4.2.2 Data Preparation . . . . .	14
4.2.3 Training of contextual models . . . . .	15
4.2.4 Model assessment and definition of context-aware guidelines	16
<b>5 State of the art</b>	17
5.1 Regression . . . . .	17
5.1.1 ARIMA . . . . .	17
5.1.2 Linear regression . . . . .	18
5.1.3 Lasso . . . . .	18
5.1.4 Ridge . . . . .	19
5.1.5 Random forest . . . . .	19

5.1.6	Gradient boosting . . . . .	20
5.2	Evaluation Metrics for Time Series Forecast . . . . .	21
5.2.1	Mean absolute error . . . . .	21
5.2.2	Mean squared error . . . . .	22
5.2.3	Root mean squared error . . . . .	22
5.2.4	$R^2$ score and adjusted $R^2$ score . . . . .	23
5.2.5	Evaluation metrics comparison . . . . .	23
<b>6</b>	<b>Methodology</b>	<b>25</b>
6.1	Data Acquisition . . . . .	25
6.2	Data Preparation . . . . .	26
6.2.1	Data cleaning . . . . .	29
6.2.2	Data integration for data enrichment . . . . .	30
6.2.3	Exploratory data analysis . . . . .	30
6.2.4	Sampling . . . . .	35
6.3	Training of contextual models . . . . .	35
6.4	Model assessment and definition of context-aware guidelines . . . . .	35
<b>7</b>	<b>Experimental analysis</b>	<b>36</b>
7.1	Seattle dataset . . . . .	36
7.1.1	ARIMA . . . . .	36
7.1.2	First experimental session . . . . .	37
7.1.3	Second experimental session . . . . .	38
7.1.4	Third experimental session . . . . .	45
7.1.5	Machine learning regression models comparison . . . . .	46
7.2	San Francisco dataset . . . . .	57
7.2.1	ARIMA . . . . .	57
7.2.2	First experimental session . . . . .	58
7.2.3	Second experimental session . . . . .	59
7.2.4	Third experimental session . . . . .	67
7.2.5	Machine learning regression models comparison . . . . .	69
<b>8</b>	<b>Conclusions and future works</b>	<b>80</b>
	<b>Bibliography</b>	<b>81</b>

# List of Tables

5.1	Evaluation metrics comparison [23]	24
6.1	Datasets analyzed	25
6.2	Weather datasets analyzed	26
6.3	Fremont Bridge statistics	26
6.4	Seattle weather statistics	27
6.5	San Francisco Bay Area statistics	27
6.6	San Francisco Bay Area weather statistics	28
7.1	ARIMA evaluation metrics Seattle	37
7.2	Random Forest metrics based on the combination of parameters	39
7.3	ARIMA evaluation metrics San Francisco	57
7.4	Random Forest metrics based on the combination of parameters	60
7.5	Evaluation metrics window size 24 horizon 6	62
7.6	Evaluation metrics window size 12 horizon 6	62
7.7	Evaluation metrics window size 6 horizon 48	63
7.8	Evaluation metrics window size 48 horizon 48	64
7.9	Evaluation metrics window size 24 horizon 6	65
7.10	Evaluation metrics window size 12 horizon 6	66
7.11	Evaluation metrics window size 6 horizon 48	66
7.12	Evaluation metrics window size 48 horizon 48	67

# List of Figures

2.1	Micro vehicles classification [3]	5
2.2	SAE 2019 [3]	7
2.3	Bike sharing in Italy [4]	8
2.4	Bike sharing in USA [5] using the number of trips as a metric	8
4.1	Proposed framework	13
4.2	Sliding window technique parameters	16
5.1	How Random Forest works	20
6.1	San Francisco data filling	29
6.2	Data enrichment San Francisco example	30
6.3	Bikes available during the considered time period	31
6.4	Bikes usage during the week	31
6.5	Seattle trends	32
6.6	Seattle rolling statistics	32
6.7	Bikes available during the considered time period	33
6.8	Bikes usage during the week	33
6.9	San Francisco trends	34
6.10	San Francisco rolling statistics	34
6.11	Bikes available resampled	35
7.1	ARIMA forecasting for Seattle dataset	36
7.2	ARIMA forecasting for Seattle dataset	37
7.3	Random Forest for Seattle dataset, using base configuration	38
7.4	Errors Random Forest with window size 12 horizon 24	40
7.5	Evaluation metrics Random Forest with window size 12 horizon 24	41
7.6	Errors Random Forest with window size 12 horizon 6	41
7.7	Evaluation metrics Random Forest with window size 12 horizon 6	41
7.8	Errors Random Forest with window size 6 horizon 48	42
7.9	Evaluation metrics Random Forest with window size 6 horizon 48	42
7.10	Errors Random Forest with window size 12 horizon 24	43



7.11	Evaluation metrics Random Forest with window size 12 horizon 24 .	43
7.12	Errors Random Forest with window size 12 horizon 6 . . . . .	43
7.13	Evaluation metrics Random Forest with window size 12 horizon 6 .	44
7.14	Errors Random Forest with window size 6 horizon 48 . . . . .	44
7.15	Evaluation metrics Random Forest with window size 6 horizon 48 .	44
7.16	Error histograms and evaluation metrics for prediction with lead time equal to 6 . . . . .	45
7.17	Error histograms and evaluation metrics for prediction with lead time equal to 12 . . . . .	45
7.18	Error histograms and evaluation metrics for prediction with lead time equal to 24 . . . . .	46
7.19	Error histograms and evaluation metrics for prediction with lead time equal to 48 . . . . .	46
7.20	Random forest with base configuration . . . . .	47
7.21	Gradient Boosting with base configuration . . . . .	48
7.22	Lasso with base configuration . . . . .	49
7.23	Linear Regression with base configuration . . . . .	50
7.24	Ridge with base configuration . . . . .	51
7.25	Random Forest with horizon 48 . . . . .	52
7.26	Gradient Boosting with horizon 48 . . . . .	53
7.27	Ridge with horizon 48 . . . . .	54
7.28	Lasso with horizon 48 . . . . .	55
7.29	Linear Regression with horizon 48 . . . . .	56
7.30	ARIMA forecasting for San Francisco dataset . . . . .	57
7.31	ARIMA forecasting for San Francisco dataset . . . . .	57
7.32	Random Forest for San Francisco dataset, prediction made using base configuration . . . . .	59
7.33	Random Forest error histograms with window size 24 horizon 6 . .	61
7.34	Random Forest error histograms with window size 12 horizon 6 . .	62
7.35	Random Forest error histograms with window size 6 horizon 48 . .	63
7.36	Random Forest error histograms with window size 48 horizon 48 . .	63
7.37	Random Forest error histograms with window size 24 horizon 6 . .	65
7.38	Random Forest error histograms with window size 12 horizon 6 . .	65
7.39	Random Forest error histograms with window size 6 horizon 48 . .	66
7.40	Random Forest error histograms with window size 48 horizon 48 . .	67
7.41	Error histograms for prediction with lead time equal to 6, plus evaluation metrics . . . . .	68
7.42	Error histograms for prediction with lead time equal to 12, plus evaluation metrics . . . . .	68
7.43	Error histograms for prediction with lead time equal to 24, plus evaluation metrics . . . . .	69

7.44	Error histograms for prediction with lead time equal to 48, plus evaluation metrics . . . . .	69
7.45	Random Forest with base configuration . . . . .	70
7.46	Gradient Boosting with base configuration . . . . .	71
7.47	Ridge with base configuration . . . . .	72
7.48	Lasso with base configuration . . . . .	73
7.49	Linear Regression with base configuration . . . . .	74
7.50	Random Forest with horizon 48 . . . . .	75
7.51	Gradient Boosting with horizon 48 . . . . .	76
7.52	Ridge with horizon 48 . . . . .	77
7.53	Lasso with horizon 48 . . . . .	78
7.54	Linear Regression with horizon 48 . . . . .	79



# Acronyms

**ML**

Machine Learning

**MAE**

Mean Absolute Error

**MSE**

Mean Squared Error

**RMSE**

Root Mean Squared Error

# Chapter 1

## Introduction

### 1.1 Context

Micromobility is a relatively new term which was coined by Horace Dediu (approximately in 2017), a Romanian-American industry analyst, who advocates that the term has in itself the definition: "micro" denotes something small or minimal and "mobility" denotes the capacity to move. He defined micro-vehicles as vehicles with a mass of maximum 1000 Kg. The intergovernmental organization called ITF tried to find a more specific definition of micro-vehicles and they arrived to the conclusion that they are vehicles with top speed equal to 45 km/h and top weight 350 kg. Finding a unique definition for micro-vehicles is difficult because there are different legislation in different countries concerning them. This is one reason why it is not so easy to find open micromobility data in every country and sometimes it is necessary to pay to have them.

This thesis aims to be able to predict as accurately as possible the bike occupancy levels of a certain area, using machine learning techniques. To do this, a framework and methodology were first created to serve as the basis for whatever dataset is to be used; mobility data need to be acquired first, then, since not all datasets that are available have the same structure and since the data needs to be arranged in a specific way in order to be utilized in the following steps, it needs to be cleaned. The following phase is "data integration for data enrichment" where the dataset in use is enriched with weather data and temporal data, then the data is explored to find the best time period to study and more important how to study the dataset. If the dataset in question has too high granularity sampling needs to be done. Then the windowing technique is applied to the dataset, the chosen Machine Learning models is trained and the prediction is done. Following these stages, a set of guidelines needs to be formulated.

There are two datasets used in this study: the first one is an open dataset based in San Francisco, specifically in the Bay Area and it is about station-based bike sharing vehicles. The second one is also an open dataset based in Seattle, specifically on the Fremont Bridge and it is about the number of bikes that cross the bridge.

The Machine Learning techniques used to perform this study are regression models such as ARIMA, Linear regression, Lasso, Ridge, Random Forest and Gradient Boosting. To evaluate the performance of these different algorithms some evaluation metrics were used such as Mean Absolute Error, Mean Squared Error, Root Mean Squared Error and  $R^2$  score. The methodology discussed earlier was applied to the selected datasets and the experimental analysis was done: first of all the ARIMA model was tested using the windowing technique, but the outcome wasn't as good as expected, then all the different techniques were tested using the same parameters and the outcome was that the Random Forest was the best model among the others. The next experimental session was done by doing all the combination between the window size and the horizon to find the best one and the prediction was done. Then the data enrichment was done by adding the time features and the prediction was compared to the first one. Then the weather features were added to compare this prediction with the two before. Lastly the unavailability of most recent data was tested by varying the parameter called "lead time".

This study could be useful for companies providing station-based sharing bicycle service to know whether there will be a need to redistribute them to other stations or to add new ones to the station under consideration, in the case of bike flow it is useful for companies or municipalities to understand whether there is enough turnout to be able to build useful services to the community in that particular area.

## 1.2 Thesis outline

This thesis has the following organization:

- **Chapter 2:** Description of micromobility and the laws concerning it in Europe and the United States and description of datasets found in the online world.
- **Chapter 3:** Description of datasets used in the study.
- **Chapter 4:** Introduction to the framework and description of the methodology used in this study.
- **Chapter 5:** Description of theoretical concepts and tools used in this study.

- **Chapter 6:** Discussion related to the application of the proposed methodology to the datasets.
- **Chapter 7:** Experimental analysis and discussion of the results obtained.
- **Chapter 8:** Conclusions and future work.

## Chapter 2

# Urban mobility data

### 2.1 Mobility and Micromobility

All types of movement within cities are referred to as "urban mobility". It encompasses forms of transportation like biking, walking, and public transportation.

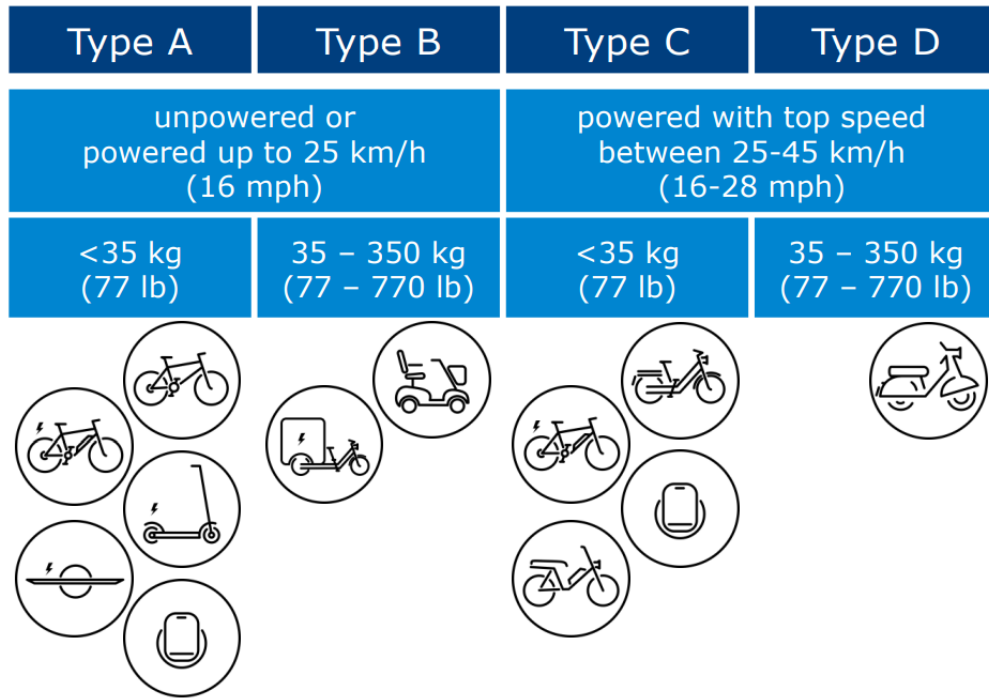
Micro-mobility, as defined by the Treccani dictionary (2020) [1], is "mobility related to short routes and distances mainly in cities, characterized by the use of means of transportation that are less heavy and bulky and potentially less polluting than traditional ones (e.g., scooters, electric scooters, pedal-assisted bicycles)."

Horace Dediu is credited with coining the term "micromobility" [2]; he maintains that it derives from the two words that make up the term: "micro" denotes something small or minimal, and "mobility" denotes the capacity to move. According to what he said earlier, the concept is "the ability of movement through minimalistic means." Dediu contends that in light of this, micromobility must be defined by focusing on its intended purpose, which is to move a human being with the least amount of interference to their freedom of movement. Weight, a characteristic of people, was the unit of measurement he used to describe this notion. He then selected a limit equal to a multiple of 5 of the average person's weight across all continents, or roughly 100 kg (taking clothing and personal goods into account). The vehicle's weight cannot exceed 500 kg, which is the limit for micromobility. To be safe, a restriction of 1000 kg was then established, which continues to exclude cars.

ITF (International Transport Forum) [3], an intergovernmental organization that serves as an advocacy group for transportation policy, claims that the phrase is vague because it encompasses a variety of light vehicles, including bicycles, kick scooters, and powered skates. They suggest that using micro vehicles, those weighing less than 350 kg and traveling at a top speed of no more than 45 km/h, is what is meant by "micromobility." Vehicles that are powered by humans and by



electricity are both included in this description. The ITF classifies micro-vehicles as follows:



**Figure 2.1:** Micro vehicles classification [3]

They are first divided into two groups based on their top speed: up to 25 km/h and between 25 and 45 km/h. They can then be further divided into two categories based on weight: less than 35 kg and between 35 and 350 kg. Although this, there are some distinctions between the United States of America and European definitions of micromobility.

### Europe

The L-category vehicles were established by European Union regulation N°168/2013 as a standard for member nations. Powered two-, three-, and four-wheel vehicles are under the L category. Depending on the power, power source, speed, length, height, and width, the classification criteria alter. A significant category known as "light two-wheel powered vehicle" is L1e, which is made up with:

- L1e-A powered bicycle: an electric bike with an auxiliary motor, a top speed of 25 km/h, and a net power range of 250 to 1000 watts.
- Two-wheel vehicles with a top speed between 26 and 45 km/h and a maximum net power of 4000 watts are classified as L1e-B two-wheel mopeds.

Excluded from this category are several micro-vehicles, such as human-powered vehicles, bicycles with pedal assistance up to 25 km/h and with an auxiliary electric motor with a maximum continuous rated power of up to 250 watts, self-balancing vehicles, and vehicles without seats.

## **United States**







Electric bikes and scooters are primarily controlled at the state level in the United States. According to state rules, an e-bike must belong to one of the following three categories in order to be used:

- A class 1 electric bike is one that has a motor that only assists when the rider is pedaling and stops once the bike achieves a speed of 32 km/h.
- A class 2 electric bicycle is one that has a motor that can only be used to move the vehicle forward and is unable to assist once it reaches a speed of 32 km/h.
- An electric bicycle classified as a class 3 is one that has a motor that only assists the rider when they are pedaling and stops when they reach a speed of 45 km/h. It also has a speedometer.

## **Other efforts to classify micromobility**

For engineers working in a variety of industries, SAE International is a U.S.-based, internationally active professional association and standards development organization. According to the J3194TM Standard, which SAE International published, powered micro mobility is a subcategory of powered vehicles that may be categorized using four primary factors (SAE, 2019):

- vehicle weight of up to 227 kg
- vehicle width of up to 1.5 m
- top speed of up to 48 km/h
- power source by an electric motor or a combustion engine

	Powered Bicycle	Powered Standing Scooter	Powered Seated Scooter	Powered Self-Balancing Board	Powered Non-Self-Balancing Board	Powered Skates
						
Center column	Y	Y	Y	Possible	N	N
Seat	Y	N	Y	N	N	N
Operable pedals	Y	N	N	N	N	N
Floorboard / foot pegs	Possible	Y	Y	Y	Y	Y
Self-balancing <sup>2</sup>	N	N	N	Y	N	Possible

**Figure 2.2:** SAE 2019 [3]

The J3194 standard identified six different categories of powered micro-vehicles: powered bicycle, powered standing scooter, powered seated scooter, powered self-balancing board, powered non-self-balancing board, and powered skates. Only automobiles intended primarily for passenger transportation and for use on paved roads and walkways are included. Only human-powered vehicles, such as standard bikes, are excluded. However it makes a distinction between three e-bike classes:

- Class 1: low-speed, pedal-assisted e-bike
- Class 2: low-speed, throttle-assisted e-bike
- Class 3: speed pedal-assisted e-bike

An alternate classification of vehicles places more emphasis on the top speed, weight, emissions, spatial footprint, and health footprint (a result of the physical activity input), normalized by the vehicle’s passenger capacity.

Apart from how micro vehicles might be categorized, they can be used privately, docked- or dockless-shared. A docked device can be picked up and released from a certain position, whereas a dockless item can be dropped and picked up from any area.

In this thesis, the focus is just on bikes.

From various reports on the web, it is possible to see how micromobility is becoming increasingly important and necessary within different cities around the world. In particular, this thesis being focused on bicycles, looking at the following graphs it can be seen how their use has grown over the years (apart during 2020 when there was the COVID-19 pandemic).

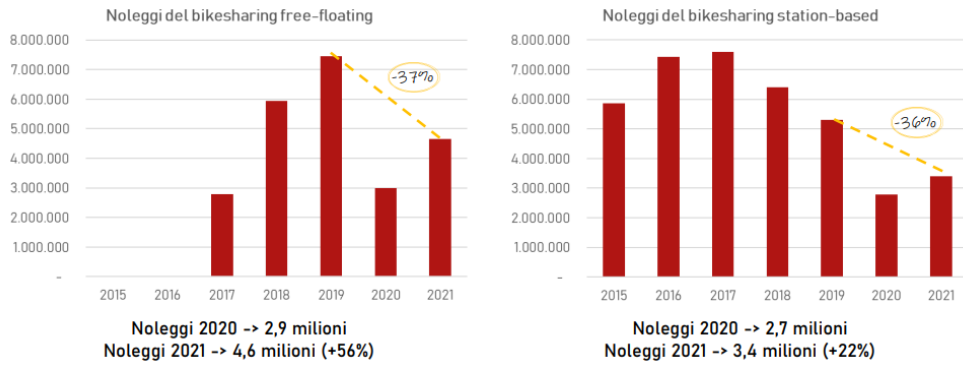


Figure 2.3: Bike sharing in Italy [4]

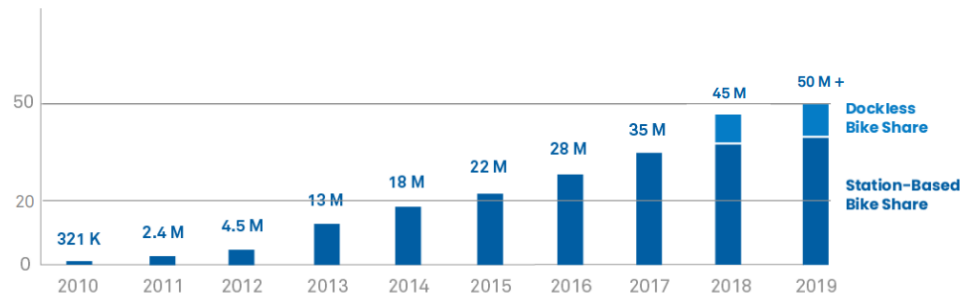


Figure 2.4: Bike sharing in USA [5] using the number of trips as a metric

There are certain open datasets available on the Internet that can be used to examine how people move around cities using sustainable vehicles (micro-vehicles), but it was quite challenging to locate accessible datasets based in Europe, particularly in Italy. This may be because shared services are very new in our nation. Some of the more pertinent datasets are given here:

- **London dataset** [6] contains data regarding e-bikes,
- **Austin dataset** [7] contains shared micromobility vehicle trip data for both e-bikes and e-scooters,
- **Berlin and Munich dataset** [8] that is a reference matrix of more than half a million e-scooters from former micromobility provider circa from the second half of 2019 for the cities of Berlin and Munich. Information on number of trips, average trip duration, and route are aggregated by the hour and in small area,

- **Chicago dataset** [9] that contains electric scooter trips taken during the 2020 Chicago pilot program,
- **Norfolk dataset** [10] that is about trips on electric scooters and electric-assisted bikes. Data represents trips from July 2019 to the present and is updated weekly.

Since the ultimate goal would be to use a dataset containing Italian data, a company called Fluctuo [11] was discovered, which was created in 2019 and is situated in France, that sells several datasets related to micromobility. The price of data varies depending on how far back in time you require them. After all this research, it is clear that data are useful and important because often it is necessary to pay to investigate mobility in various cities and nations.

# Chapter 3

## Data description

This chapter describes the datasets utilized in this thesis.

For this study anonymous mobility data from the bike and e-scooter rental business was used.

### 3.1 Station-based SF Bay Area Bike Share

The San Francisco Bay Area Bike Share [12] was the first dataset taken into consideration and it is about station-based bike sharing vehicles. This dataset is made up of four files, each of which contains details about various aspects of renting bicycles. The following files are included: `station.csv` (which contains data that represents a station where users can pick up or return bikes), `status.csv` (which contains data about the number of bikes and docks available for a given station and minute), `trips.csv` (which contains data about individual bike trips), and `weather.csv` (which contains data about the weather on a specific day for specific zip codes). Only weather records and statistics regarding the number of bikes at each station were taken into account for this study.

The file "`status.csv`" consists of the following fields:

- **station\_id**: this is the identification number of the specific station.
- **bikes\_available**: this number represents the number of bikes that are parked in a specific station at a specific time.
- **docks\_available**: this number represents the number of docks that are free in a specific station at a specific time.
- **time**: this field in the data set refers to the time when the data on the number of bikes available/docks available were collected.

The "**weather.csv**" file contains information about the weather and includes the following fields:

- **date**: date in which the weather data was collected.
- **max\_temperature\_f**: maximum temperature expressed in Fahrenheit.
- **mean\_temperature\_f**: mean temperature expressed in Fahrenheit.
- **min\_temperature\_f**: minimum temperature expressed in Fahrenheit.
- **max\_dew\_point\_f**: maximum dew point expressed in Fahrenheit.
- **mean\_dew\_point\_f**: mean dew point expressed in Fahrenheit.
- **min\_dew\_point\_f**: minimum dew point expressed in Fahrenheit.
- **max\_humidity**: maximum humidity.
- **mean\_humidity**: mean humidity.
- **min\_humidity**: minimum humidity.
- **max\_sea\_level\_pressure\_inches**: maximum sea level pressure expressed in inches.
- **mean\_sea\_level\_pressure\_inches**: mean sea level pressure expressed in inches.
- **min\_sea\_level\_pressure\_inches**: minimum sea level pressure expressed in inches.
- **max\_visibility\_miles**: maximum visibility expressed in miles.
- **mean\_visibility\_miles**: mean visibility expressed in miles.
- **min\_visibility\_miles**: minimum visibility expressed in miles.
- **max\_wind\_Speed\_mph**: maximum wind speed expressed in mph.
- **mean\_wind\_speed\_mph**: mean wind speed expressed in mph.
- **max\_gust\_speed\_mph**: maximum gust speed expressed in mph.
- **precipitation\_inches**: precipitation during that date expressed in inches.
- **cloud\_cover**: cloud cover expressed in oktas.

- **events**: events happened during that date, which could be: nan, 'Fog', 'Rain', 'Fog-Rain', 'rain', 'Rain-Thunderstorm'.
- **wind\_dir\_degrees**: wind direction degrees.
- **zip\_code**: zip code that expresses where these data are valid.

## 3.2 Seattle: Fremont Bridge Bicycle Counter

The Fremont Bridge Bicycle Counter dataset [13] is based on a counter that tracked the number of bicycles utilizing the pedestrian and cycling lanes to cross the bridge between October 2012 and March 2023.

This data set is characterized by:

- **Date**: Date in which the number of bikes is recorded.
- **Fremont Bridge Sidewalks, south of N 34th St**: total number of bikes that come on the bridge from west and east sidewalks.
- **Fremont Bridge Sidewalks, south of N 34th St Cyclist East Sidewalk**: number of bikes from east sidewalk.
- **Fremont Bridge Sidewalks, south of N 34th St Cyclist West Sidewalk**: number of bikes from west sidewalk.

Since the previous data collection didn't include weather information, a new dataset was needed. The chosen one is called "Did it rain in Seattle?" [14] and. The rain sensor over Seattle's airport provided the data used here, thus even though it is not over the bridge, it was chosen to be utilized.

The dataset that was selected is made up of the following:

- **DATE**: Date when the weather was observed.
- **PRCP**: precipitation expressed in inches.
- **TMAX**: maximum temperature expressed in Fahrenheit.
- **TMIN**: minimum temperature expressed in Fahrenheit.
- **RAIN**: boolean that expresses if it rained or not.



# Chapter 4

## Data analysis framework and methodology

This chapter will go through the study’s methodology and the framework that was employed.

### 4.1 Framework

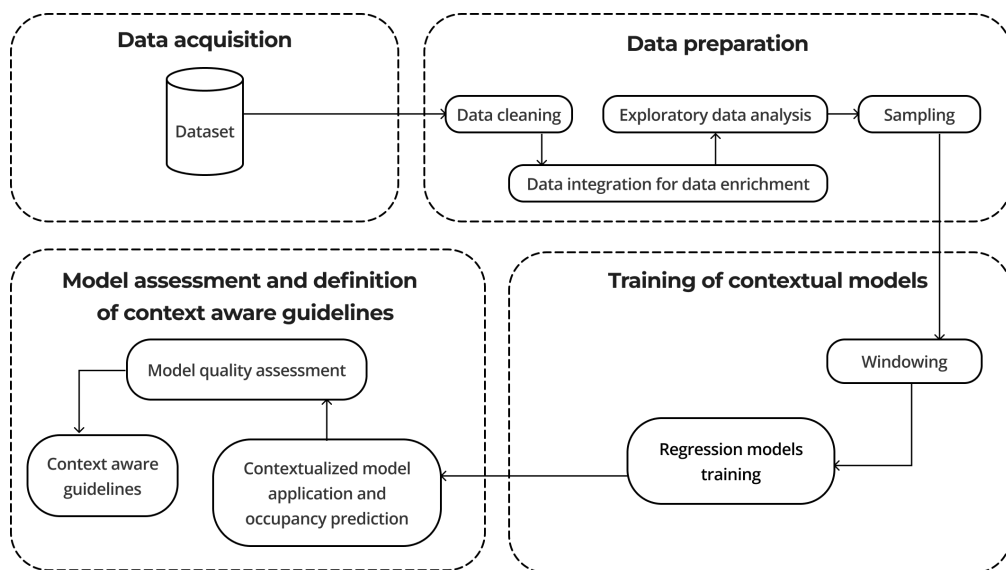


Figure 4.1: Proposed framework

- **Data acquisition:** E-bikes rental data and the related weather are acquired and stored.

- **Data preparation:** In this stage, a new, cleaned, enriched, and sampled dataset was created by selecting a consistent time period for the data, enriching the dataset with contextual descriptors (such as temporal information and meteorological data), sampling at a selected time rate, and filling in the gaps with interpolation methods.
- **Training of contextual models:** prediction models are trained on the considered data set using the windowing technique. Models are Machine Learning-based algorithms.
- **Model assessment and definition of context-aware guidelines:** A test sample is used to evaluate the effectiveness of the top machine learning model in predicting the occupancy levels in the near future. The system managers could be given a set of common guidelines based on the results attained in various circumstances.

## 4.2 Proposed Methodology

### 4.2.1 Data Acquisition

The initial stage of the process is data acquisition, during which data is collected from multiple open sources.

### 4.2.2 Data Preparation

When the first step is completed and all the necessary data are present, they must be processed before moving on to analyze them. The four steps which make up this phase are data cleaning, data integration for data enrichment, exploratory data analysis, and sampling.

#### **Data cleaning**

Data collected in the initial stage must first be cleaned up and transformed into structured data before it can be utilized for any analysis or modeling. If data isn't stored properly, it may be prone to mistakes, which might easily sabotage the study or the program might not work. The data must be evaluated and errors must be sought in order to arrive at more precise conclusions. The most frequent errors that might be encountered are:

- Missing values
- Duplicate values

- Values set to null when they should be zero or the opposite
- Time zone differences
- Corrupted values (i.e. invalid entries)
- Date range errors

Also, Machine Learning models need numerical values to work, so, one-hot encoding needs to be used on the categorical values to convert them into numerical ones.

### **Data integration for data enrichment**

This stage involves integrating data from many open sources to expand the concerned dataset as well as define the context.

### **Exploratory data analysis**

This preliminary data exploration may help generate testable hypotheses and determine which subsets of data to employ for subsequent modeling. In this stage, the available data is examined for any hidden patterns. Additionally, examining the numerous impact factors on the target variable as well as their strength is attempted. These answers, as well as how the separate traits relate to one another and what may be done to get the desired outcomes, can be obtained from this procedure. This offers a starting point for the modeling process as well.

### **Sampling**

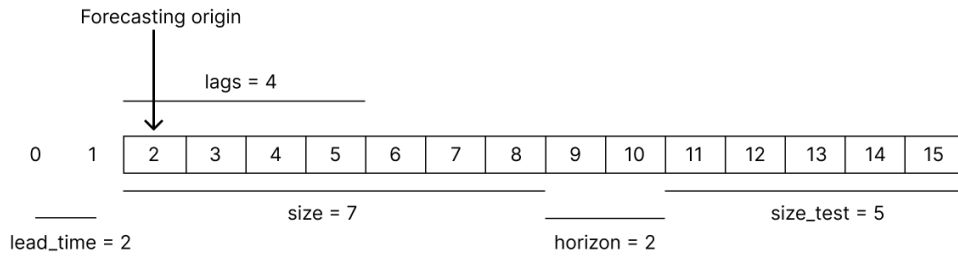
In this phase data sampling is used to select, modify, and analyze a representative selection of data points. Typically, it is employed when working with extremely large datasets.

## **4.2.3 Training of contextual models**

To be able to provide an accurate prediction, the selected models must be trained. The properties of the sliding window algorithm, which we choose to utilize when training the models, are displayed in the following section.

### **Windowing**

The image below summarizes and explains the different parameters of the sliding window technique.



**Figure 4.2:** Sliding window technique parameters

- **lags:** represents how many previous steps will be considered, essentially it's the window size.
- **lead\_time:** represents the latency between the origin of forecasting and fetched data.
- **horizon:** represents the distance between the training set and the test set.
- **size:** represents the size of the training set.
- **size\_test:** represents the size of the test set.

### Regression models training

The occupancy levels and contextual information, such as weather and temporal information, are taken into account as input variables in the regression models. The data from the sliding window training set were used to train the selected models.

#### 4.2.4 Model assessment and definition of context-aware guidelines

System managers might develop context-aware guidelines for customizing the settings and use of the intelligent system to the context of the real world based on the results achieved. Guidelines may indicate:

- The best prediction model in predicting future occupancy levels.
- The ranges within which the different parameters of the sliding window technique must lie in order to achieve accurate prediction.

# Chapter 5

## State of the art

This chapter presents the theoretical concepts underlying this thesis.

### 5.1 Regression

Regression [15] is a form of supervised learning method used in ML to forecast continuous numerical values based on input variables. A link between the dependent variable (also called the goal or outcome variable) and one or more independent variables (sometimes called predictors or features) is intended to be established using this statistical modeling technique. Finding a mathematical function that accurately captures the relationship between the input variables and the target variable is the aim of regression analysis. Then, using fresh, unforeseen data, this function can be utilized to create predictions.

This section will cover the primary supervised learning regression models used in this study. They are: ARIMA, Linear Regression, Lasso, Ridge, Random Forest, and Gradient Boosting, presented in that sequence.

#### 5.1.1 ARIMA

ARIMA[16] stands for "autoregressive integrated moving average" and it is a forecasting algorithm. In ARIMA regression, the dependent variable (target variable) is represented as a linear combination of its own historical past values, the historical past values of the errors (also called residuals), and potentially some outside variables. An ARIMA model is defined by the three parameters  $p$ ,  $d$ , and  $q$  that determine, respectively, the orders of autoregressive (AR), differencing (I) and the moving average (MA) components:

- $p$  is the number of autoregressive terms, it refers to the number of lagged terms to be used as predictors;

- $d$  is the minimum number of differencing needed to make the series stationary (if  $d = 0$ , the time series is already stationary);
- $q$  is the number of moving averages, it refers to the number of lagged forecast errors that should be included into the model;

### 5.1.2 Linear regression

The simplest linear regression model [15] is the one with only one independent variable and the equation is the following:  $y = mx + b$ , where  $y$  is the target variable,  $x$  is the independent variable,  $m$  represents the slope of the line,  $b$  is the y-intercept (the value of the target variable when  $x$  is 0). To reduce the discrepancy between the projected values of  $y$  and the actual values of  $y$ , the linear regression seeks to estimate the values of  $m$  and  $b$ . This is accomplished using the least square method, which minimizes the sum of the squared disparities between the predicted and actual values.

Linear regression can be extended to multiple independent variables and in this case it involves a linear combination of the input variables:

$$y = b_0 + b_1x_1 + \dots + b_n * x_n \quad (5.1)$$

where  $x_1 \dots x_n$  are the independent variables,  $b_0$  is the y-intercept,  $b_1 \dots b_n$  are the coefficients associated with each independent variable. This model is a linear function of the parameters  $(b_0, \dots, b_n)$ , but also a linear function of the input variables  $(x_1, \dots, x_n)$ .

### 5.1.3 Lasso

Lasso [17] is an acronym that stands for "Least Absolute Shrinkage and Selection Operator" and it is also known as L1 regularization. It is a method for linear regression which utilizes regularization by including a penalty term in the loss function. This term is proportional to the sum of the absolute values of the coefficients, multiplied by a tuning parameter called lambda ( $\lambda$ ):

$$Cost(W) = RSS(W) + \lambda * (Sum\ of\ the\ absolute\ weights) \quad (5.2)$$

Where:

- RSS represents the residual sum of squares.
- $\lambda$  represents the amount of shrinkage of coefficients (larger values lead to more shrinkage, smaller values leads to lesser shrinkage).

The addition of this penalty term results in an automatic feature selection. So, this model performs both regression and feature selection.

### 5.1.4 Ridge

Ridge regression [17] is a regularization technique (uses L2 regularization). By including a penalty component to the cost function, it expands the ordinary least squares (OLS) regression technique. The objective is to identify the regression coefficients that minimize the squared difference between the predicted and actual values. It lessens the size of the coefficients thanks to the additional penalty term. This term is proportional to the squared sum of the coefficients multiplied by a hyperparameter called lambda ( $\lambda$ ) or the regularization parameter.

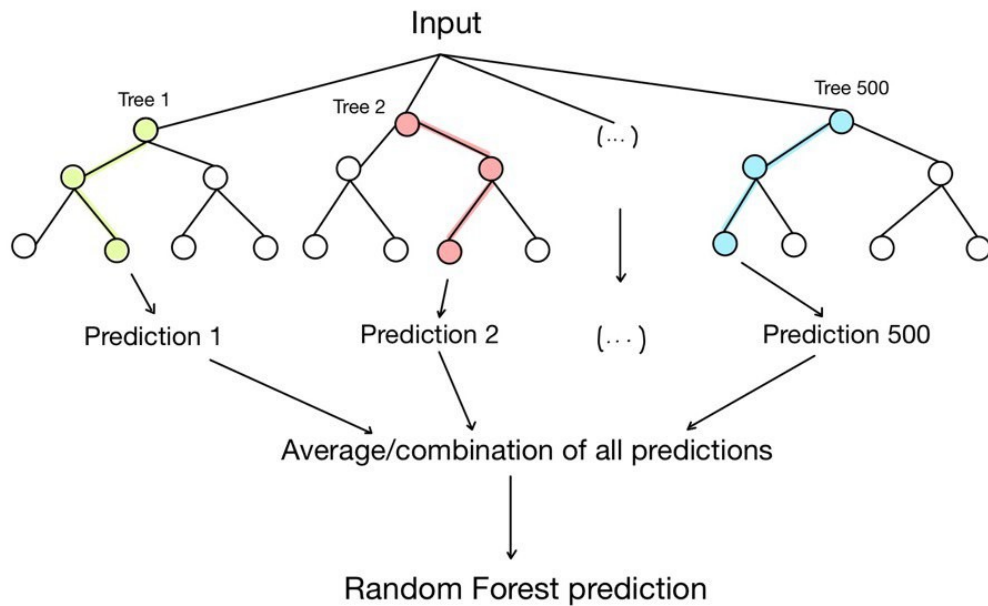
$$Cost(W) = RSS(W) + \lambda * (Sum\ of\ the\ squared\ coefficients) \quad (5.3)$$

Where:

- RSS represents the residual sum of squares, measures the error between the predicted values and the actual values.
- $\lambda$  represents the amount of shrinkage of coefficients (larger values lead to more shrinkage, smaller values leads to lesser shrinkage).

### 5.1.5 Random forest

Random Forest Regression [18] is an algorithm that uses ensemble learning method for regression: it creates an ensemble of decision trees and each tree is trained on a different subset of the data. During training, the random forest model simultaneously builds a huge number of trees and generates a class that represents the mean of all the individual trees. The unpredictability added throughout the tree-building process is what is meant by the "random" aspect of this algorithm.



**Figure 5.1:** How Random Forest works

This is how the Random Forest algorithm works:

1. Randomly picks subsets of data (that can contain duplicate samples) from the training set.
2. Builds a decision tree associated to each subset of data. A split is created based on a feature that best separates the data based on a particular criterion at each node of the tree.
3. Once all the trees have been built, each one makes forecasts on its own. To arrive at the final prediction, the anticipated values of each tree are averaged or combined.
4. Predictions on unseen data can now be made using the trained model.

### 5.1.6 Gradient boosting

Gradient boosting [19] is a machine learning algorithm which works on the ensemble technique called 'Boosting'. The term "gradient boosting" refers to the optimization process used to iteratively improve the model's performance. Its idea is to train weak learners (decision trees) sequentially, each trying to correct its predecessor. This iterative process enables the model to improve its predictions by learning from its mistakes.



These are the steps performed by a Gradient Boosting algorithm:

1. Initialize the model with a simple predictive model, it could be a base tree with single root node. It is the initial guess for all the samples.
2. Calculate the residuals (differences between predicted and actual target values) for each training example.
3. Usually a decision tree is fit to the residuals. The aim is to find the best split points that will reduce the residuals.
4. The tree is scaled by learning rate (value between 0 and 1). This learning rate determines the contribution of the tree in the prediction.
5. This new tree is added to the ensemble to predict the result and the previous step are repeated until a stopping criterion is met (maximum number of trees is achieved, new trees don't improve the fit etc...).

Once all the iterations are completed, the final prediction model is the combination of all the trees in the ensemble.

## 5.2 Evaluation Metrics for Time Series Forecast

Evaluation metrics [20] offer objective measurements of how well a model is performing. We used scale-dependent metrics, which means that they are expressed in the units of the data considered. All these metrics were used to assess which regression model and which parameters were the best to make the prediction as good as possible.

### 5.2.1 Mean absolute error

MAE [20] calculates the average absolute difference between between the predicted and actual data. MAE assigns each error the same weight. Its formula is the following:

$$MAE = \frac{1}{n} \sum_{j=1}^n |e_j| \quad (5.4)$$

where

- $n$  is size of the data set,
- $e_j$  is the difference between  $A_j$  and  $P_j$
- $A_j$  are the actual values
- $P_j$  are the predicted values.

### 5.2.2 Mean squared error

MSE [20] calculates the average squared difference between the predicted and actual values. Compared to MAE, MSE gives greater errors more weight. Its formula is the following:

$$MSE = \frac{1}{n} \sum_{j=1}^n e_j^2 \quad (5.5)$$

where

- $n$  is size of the data set,
- $e_j$  is the difference between  $A_j$  and  $P_j$
- $A_j$  are the actual values
- $P_j$  are the predicted values.

### 5.2.3 Root mean squared error

RMSE [20] calculates the average difference in magnitude between expected values and actual values, known as residuals or prediction errors. It is an extension of the MSE, it makes the error metric more understandable.

Its formula is as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n e_j^2} \quad (5.6)$$

or

$$RMSE = \sqrt{MSE} \quad (5.7)$$

where

- $n$  is size of the data set,
- $e_j$  is the difference between  $A_j$  and  $P_j$
- $A_j$  are the actual values
- $P_j$  are the predicted values.

### 5.2.4 $R^2$ score and adjusted $R^2$ score

The performance of the model considered can be showed thanks to the metric called  $R^2$  score [21]. Its value, ideally, ranges from 0 to 1. Its formula is the following:

$$R^2 = 1 - \frac{RSS}{TSS} \quad (5.8)$$

where:

- RSS is the sum of squares of residuals
- TSS is the total sum of squares

Adjusted  $R^2$  [22] is a modified version of the  $R^2$  and it depends on the features that are added to the model: if the new feature improves the model more than expected the value of it increases, instead if the new feature improves the model less then expected the value of it decreases. Its formula is the following:

$$AdjR^2 = 1 - \frac{\frac{RSS}{(n-K)}}{\frac{TSS}{n-1}} \quad (5.9)$$

where:

- K is the number of parameters fit
- n is the number of data points

### 5.2.5 Evaluation metrics comparison

The table presented below summarizes the advantages and disadvantages of the different evaluation metrics examined above.

Evaluation metric	Advantages	Disadvantages
MAE	MAE is computationally cheap because of its simplicity and provides an even measure of how well the model performs. MAE is less sensitive towards outliers	MAE follows a linear scoring approach, which means that all errors are weighted equally when computing the mean. Because of the steepness of MAE, we may hop beyond the minima during back propagation. MAE is not differentiable at zero, therefore it can be challenging to compute gradients.
MSE	MSE aids in the efficient convergence to minima for tiny mistakes as the gradient gradually decreases. MSE values are expressed in quadratic equations, aids to penalizing model in case of outliers.	Squaring the values accelerates the rate of training, but a higher loss value may result in a substantial leap during back propagation, which is undesirable. MSE is especially sensitive to outliers, which means that significant outliers in data may influence our model performance.
RMSE	RMSE works as a training heuristic for models. Many optimization methods choose it because it is easily differentiable and computationally straightforward. Even with larger values, there are fewer extreme losses, and the square root causes RMSE to penalize errors less than MSE.	As RMSE is still a linear scoring function, the gradient is abrupt around minima. The scale of data determines the RMSE, as the errors' magnitude grows, so does sensitivity to outliers. In order to converge the model the sensitivity must be reduced, leading to extra overhead to use RMSE.

**Table 5.1:** Evaluation metrics comparison [23]

# Chapter 6

## Methodology

This chapter reports the results obtained by applying the methodology shown in Chapter 4 to the datasets that were chosen.

### 6.1 Data Acquisition

To study bike occupancy for a certain station and the flow of bikes passing through a certain area, bike usage data using different datasets was acquired and stored. It was decided to focus only on the occupancy or presence of bicycles, so the data related to the actual rental, such as the start time of the trip or the duration of the trip, was excluded.

The San Francisco dataset contains data covering the time period starting from August 2013 up to August 2015.

The Fremont Bridge dataset contains data covering the time period starting from 10th March 2012 up to 30th September 2022.

The Seattle weather dataset contains data covering the time period starting from 1st January 1948 up to 14th December 2017.

Dataset name	number of samples	time period
Fremont Bridge	87600	3649 days 23:00:00
San Francisco Bay Area	71984434	692 days 23:59:00

**Table 6.1:** Datasets analyzed

---

Dataset name	number of samples	time period
Did it rain in Seattle?	25551	25550 days 00:00:00
San Francisco Bay Area	3665	732 days 00:00:00

---

**Table 6.2:** Weather datasets analyzed

## 6.2 Data Preparation

During this stage the data were processed so that they could be used for further analysis. To achieve this, data need to be analyzed to take only relevant information into account. Having two different datasets available, the same procedure was applied with a few caveats.

### Seattle dataset

	bikes_available
total entries	87600
max value	1097
min value	0
mean value	107.24084899413148
non-null entries	87586

**Table 6.3:** Fremont Bridge statistics

It can be observed that the Seattle dataset includes null values. These anomalies must be fixed in order to analyze these data.

### Seattle weather dataset

RAIN	
total entries	25551
non-null entries	25548

**Table 6.4:** Seattle weather statistics

It can be observed that the Seattle weather dataset includes null values. These anomalies must be fixed in order to analyze these data.

### San Francisco dataset

bikes_ available	
total entries	260491
max value	24
min value	0
mean value	13.15929149183657
non-null entries	260491

**Table 6.5:** San Francisco Bay Area statistics

Once the San Francisco dataset was examined, it could be seen that it should have data for each minute that has passed; however, not all minutes are present, which could lead to issues that need to be resolved.

San Francisco weather dataset

	total entries	non-null entries
max_temperature_f	3665	3661
mean_temperature_f	3665	3661
min_temperature_f	3665	3661
max_dew_point_f	3665	3661
mean_dew_point_f	3665	3661
min_dew_point_f	3665	3661
max_humidity	3665	3661
mean_humidity	3665	3661
min_humidity	3665	3661
max_sea_level_pressure_inches	3665	3664
mean_sea_level_pressure_inches	3665	3664
min_sea_level_pressure_inches	3665	3664
max_visibility_miles	3665	3652
mean_visibility_miles	3665	3652
min_visibility_miles	3665	3652
max_wind_Speed_mph	3665	3664
mean_wind_Speed_mph	3665	3664
precipitation_inches	3665	3664
max_gust_speed_mph	3665	2766
cloud_cover	3665	3664
events	3665	522
wind_dir_degrees	3665	3664
zip_code	3665	3665

**Table 6.6:** San Francisco Bay Area weather statistics



It can be observed that most of the features inside the Seattle weather dataset include null values, except the "date" and the "zip\_code" ones. These anomalies must be fixed in order to analyze these data.

### 6.2.1 Data cleaning

For all datasets used in this study, only a portion of time was chosen to be considered in order to better study the phenomenon and avoid some of the problems mentioned above.

#### Seattle dataset

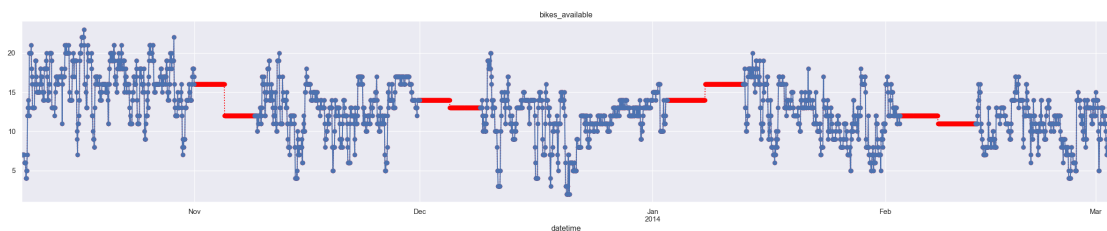
In this dataset, the field name "Fremont Bridge Sidewalks, south of N 34th S" was renamed to "bikes\_available" and a feature selection was done, because the other features were not useful for the purpose of this thesis.

#### Seattle weather dataset

This dataset did not contain any null or missing values; rather, the "RAIN" field's data type had been modified from true/false to 1/0 and the dataset was filtered to be joined with the bike flow dataset.

#### San Francisco dataset

In order to simplify the study of this phenomenon, a particular station was selected and its situation over time was studied. The "nearest neighbor" interpolation approach to fill in the missing values was used: the values nearest to the point we're aiming to fill are those that are used in this approach to fill the empty area.



**Figure 6.1:** San Francisco data filling

#### San Francisco weather dataset

The "date" field was changed to a date type object, and the "events" field, which listed the many kinds of weather occurrences that happened on that specific day, was

transformed to its internal fields with the values 0/1 to denote the presence or absence of that specific phenomenon. The feature that displayed the amount of precipitation in inches was changed to a numeric type. In addition, null or missing values were present for many fields in the dataset, these were replaced with the value of the average of all field values. In order to have the weather data for the chosen station, it was also filtered against the zip code. Although this dataset had a large number of features, not all of them were pertinent to our investigation, hence a feature selection was made.

### 6.2.2 Data integration for data enrichment

During this phase calendar descriptors were extracted from the datetime index such as date, weekday, hour, month, weekend and holiday/working day. Also meteorological data, associated to the day that was considered in our prediction, was extracted from the weather dataset and joined with the main dataset. All this information is important because it helps capture trends in the analyzed data.

	bikes_available	hour_sin	hour_cos	month_sin	month_cos	weekday_sin	weekday_cos	is_weekend
<b>datetime</b>								
2013-01-09 00:00:00	11	0.000000e+00	1.0	5.000000e-01	0.866025	0.866025	-0.5	0
2013-01-09 00:01:00	11	0.000000e+00	1.0	5.000000e-01	0.866025	0.866025	-0.5	0
2013-01-09 00:02:00	11	0.000000e+00	1.0	5.000000e-01	0.866025	0.866025	-0.5	0
2013-01-09 00:03:00	11	0.000000e+00	1.0	5.000000e-01	0.866025	0.866025	-0.5	0
2013-01-09 00:04:00	11	0.000000e+00	1.0	5.000000e-01	0.866025	0.866025	-0.5	0
...	...	...	...	...	...	...	...	...

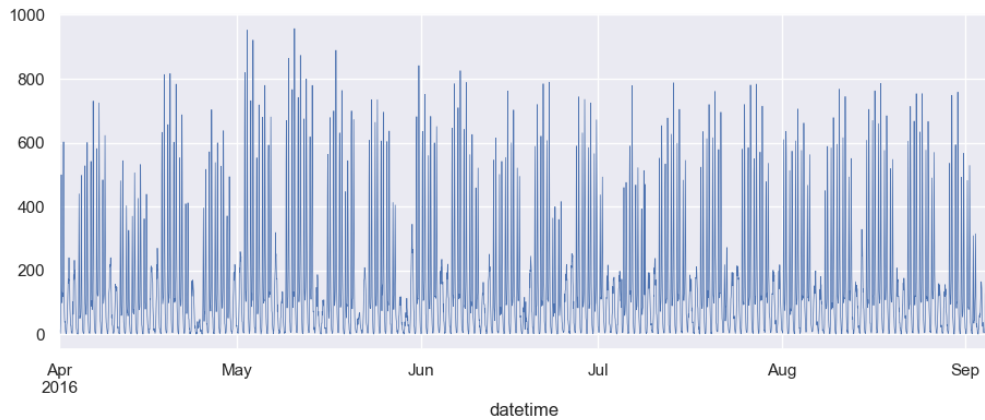
Figure 6.2: Data enrichment San Francisco example

### 6.2.3 Exploratory data analysis

To investigate the data and assess its trends in this step, a consistent and interesting time period must be selected.

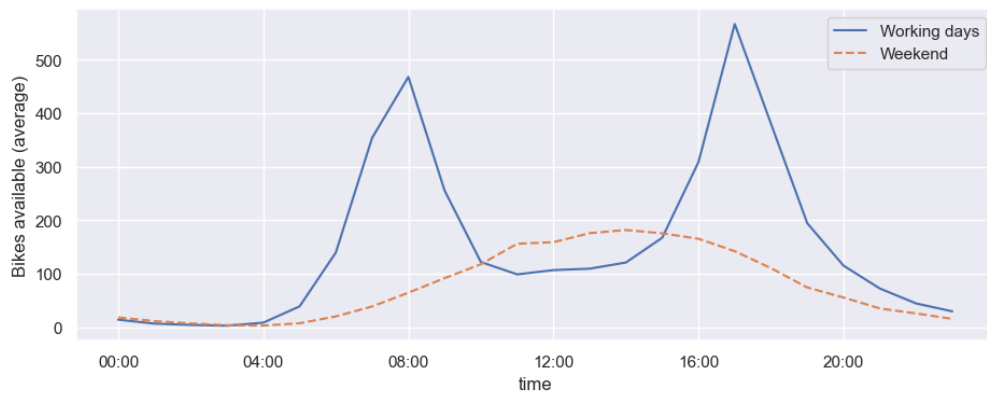
#### Seattle dataset

It was decided to take into account the time period that goes from the 1st April of 2016 at 00:00:00 to the 4th of September of 2016 at 23:00:00 (6.3).



**Figure 6.3:** Bikes available during the considered time period

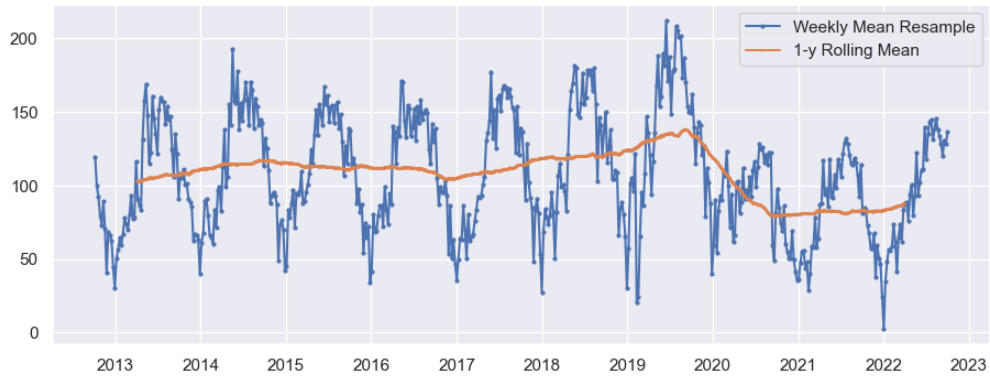
The usage of bikes during the week has been shown to be relevant, so, exploiting the properties of the index, the amount of bikes that were crossing the bridge against the time considering weekdays and working days was plotted (6.4).



**Figure 6.4:** Bikes usage during the week

From the plot above (6.4), it is clear that most individuals use the Fremont Bridge to go to work or come back home from work, as the peak bike traffic hours are around 8 AM and 5 PM. As opposed to weekdays, there is much less traffic on the bridge during weekends.

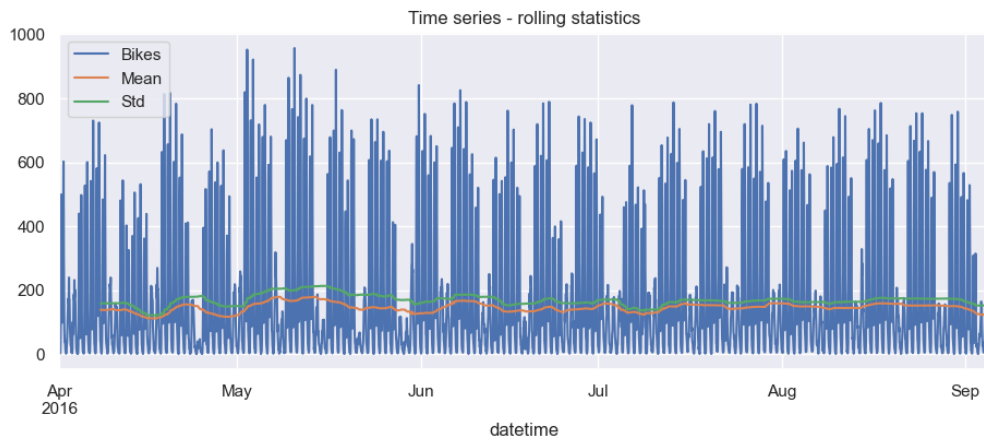
The weekly mean of the number of bikes and the one-year rolling mean were plotted using the entire dataset in order to better understand the trends seen in this dataset (6.5).



**Figure 6.5:** Seattle trends

From the plot above it can be seen that the bikes were used mainly during the central parts of the year. However, starting in 2020, when the COVID-19 epidemic began, there is a decline in usage.

Then, a consistent time period was chosen and, to see how the dataset behaves, the actual number of bikes was plotted along with the mean and the standard deviation (6.6).

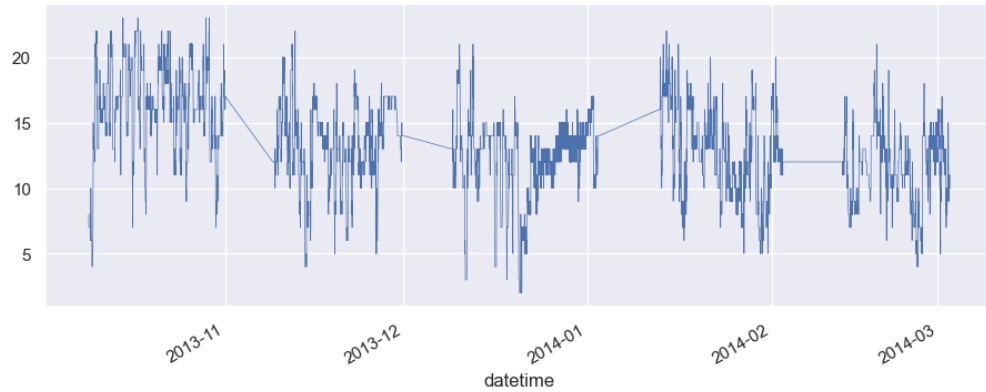


**Figure 6.6:** Seattle rolling statistics

From the plot above it can be seen that Both the mean and the standard deviation, which varies little from the mean itself, change little over time. Because the measurements made on different days of the month have such wide variations in values, the standard deviation is higher than the mean.

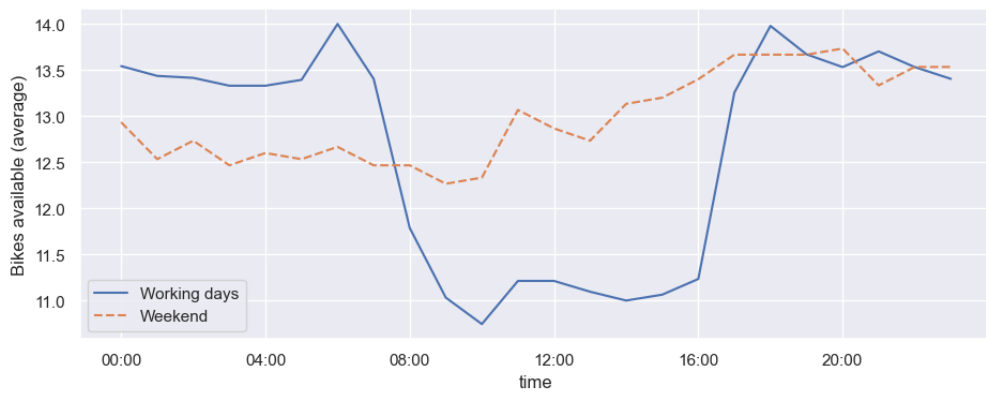
### San Francisco dataset

The time period that goes from the 1st October 2013 at 00:00:00 to 4th March 2014 at 23:00:00 was taken into account (6.7).



**Figure 6.7:** Bikes available during the considered time period

The usage of bikes during the week has been shown to be relevant, so, exploiting the properties of the index, the amount of bikes that were crossing the bridge against the time considering weekdays and working days was plotted (6.8).

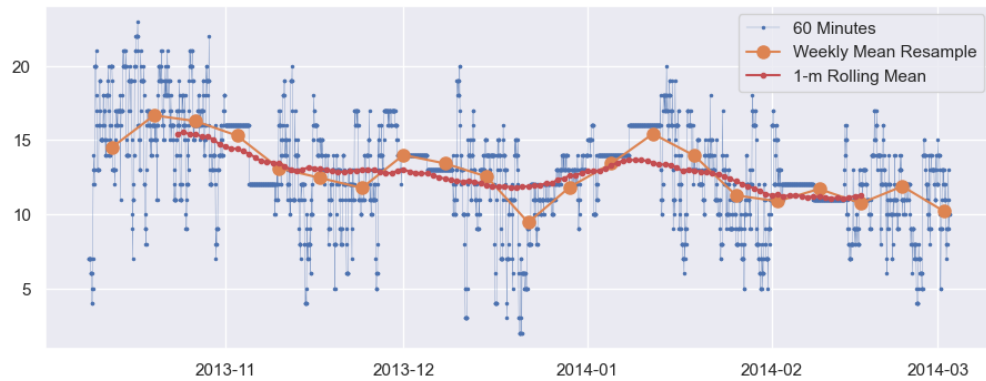


**Figure 6.8:** Bikes usage during the week

From this distribution (6.8) it can be seen that people living in San Francisco use bikes mainly during weekdays during working hours (from more or less 9 AM to 4 PM).

It was important to examine how the weekly mean and 1-month rolling mean contrasted to the re-sampled dataset 6.9. (devo cambiare le immagini con il nuovo

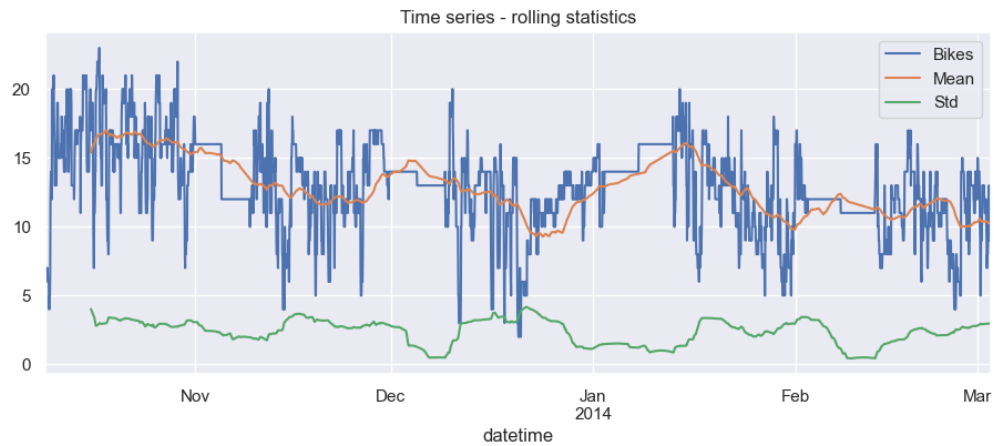
sampling)



**Figure 6.9:** San Francisco trends

From the plot above (6.9) it can be seen that both the weekly mean and the monthly rolling mean vary quite a bit.

Then, a consistent time period was chosen and, to see how the dataset behaves, the actual number of bikes was plotted along with the mean and the standard deviation (6.10).



**Figure 6.10:** San Francisco rolling statistics

From the plot above (6.10) it can be seen that both the mean and the standard deviation vary quite a bit over time. Focusing on the standard deviation curve, it can be said that its value consistently falls between 0 and 5, which denotes low data variability.

## 6.2.4 Sampling

### Seattle dataset

This dataset was resampled with weekly frequency and with daily frequency just to visualize how the dataset behaves, but no sampling was used during forecasting.

### San Francisco dataset

This dataset was resampled with 60 minutes frequency, because the granularity was too high.



Figure 6.11: Bikes available resampled

## 6.3 Training of contextual models

See Chapter 7.

## 6.4 Model assessment and definition of context-aware guidelines

Given the analysis completed and the experiments carried out, it can be inferred that the best machine learning regression model to use in this context is the Random Forest and the sliding window's parameters should be set as follows in order to have a good prediction: the window size should be 12 (hours) or 24 (hours), and the horizon should be between 6 (hours) and 48 (hours).

# Chapter 7

## Experimental analysis

This chapter will describe the steps performed on the San Francisco dataset and Seattle dataset to achieve the best possible prediction. These experiments were conducted by varying the parameters that characterize the sliding window technique and carried out using an ideal configuration of the algorithms, which was achieved by fine-tuning the hyperparameters using a grid search.

### 7.1 Seattle dataset

#### 7.1.1 ARIMA

To test how well the ARIMA model works with our dataset, the first ever experimental session was run using this model with the sliding window technique.

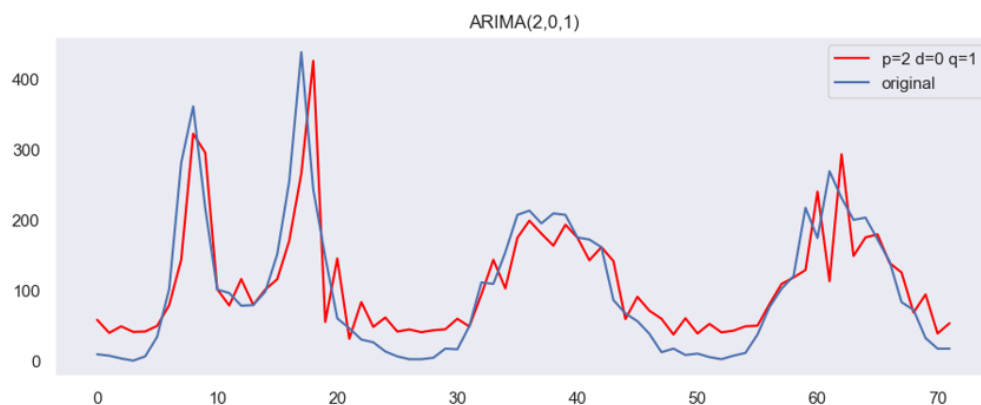
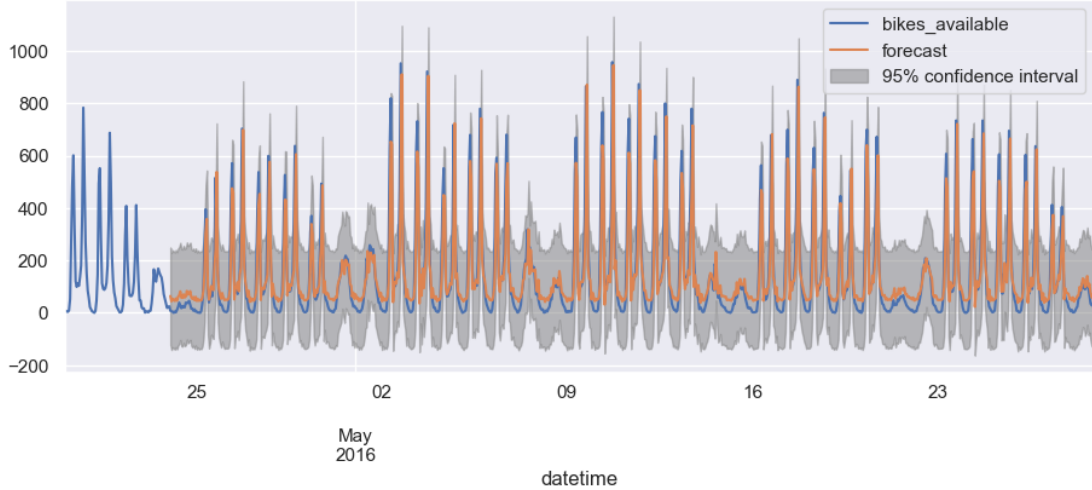


Figure 7.1: ARIMA forecasting for Seattle dataset



MAE	MSE	$R^2$	MAPE
40.508	3022.507	0.675	2548398381496615.500

**Table 7.1:** ARIMA evaluation metrics Seattle



**Figure 7.2:** ARIMA forecasting for Seattle dataset

According to the plots and the table above, even though the forecast curve resembles the curve of actual values, this model does not predict very well: the  $R^2$  score is quite low, the MSE is quite high, which indicates that data points are dispersed widely around the mean of our dataset, increasing the error. However, because our values range from approximately 0 to approximately 900, the MAE is not too bad. All things considered, this model does not perform particularly well.

### 7.1.2 First experimental session

In order to determine which regressor could most accurately predict the number of bikes crossing the Fremont Bridge for the Seattle dataset, the first round of experiments involved applying each regressor to the dataset in question. To do this, the horizon parameter was changed, and the other parameters were configured using a base configuration: lags (window size) equal to 6 (hours), lead\_time equal to 1, size equal to 24\*120, size\_test equal to 24.



**Figure 7.3:** Random Forest for Seattle dataset, using base configuration

Comparing each output obtained from the application of the regressors, due to the evaluation metrics considered (RMSE, MSE, MAE,  $R^2$ ), the Random Forest was found to be the best regressor among all, having each metric better than all other regressors considered.

### 7.1.3 Second experimental session

Since Random Forest turned out to be the best regression model, in the second round of experiments we focused on finding the best parameters for the sliding window technique using this regression model. For this round of experiments we decided to use as features only the instants that make up the sliding window, for example if the window size is equal to 6 hours, the features taken into account will be the current hour plus the previous 5 hours taken individually.

Window size	horizon	MAE	RMSE	MSE	$R^2$
6	6	21.12	36.92	13613.08	0.971775
6	12	21.20	37.56	1410.48	0.970747
6	24	21.09	37.68	1419.54	0.970480
6	48	21.37	38.45	1478.33	0.969896
<b>12</b>	<b>6</b>	<b>15.59</b>	<b>24.76</b>	<b>613.14</b>	<b>0.987297</b>
12	12	15.93	25.25	637.51	0.986776
<b>12</b>	<b>24</b>	<b>15.72</b>	<b>24.60</b>	<b>605.36</b>	<b>0.987410</b>
12	48	15.91	25.34	642.18	0.986924
24	6	15.66	26.38	695.89	0.985585
24	12	15.71	26.74	714.94	0.985175
24	24	15.87	27.88	777.11	0.983864
24	48	15.76	27.14	736.63	0.985009
48	6	16.41	28.41	806.99	0.983293
48	12	16.19	28.14	792.12	0.983573
48	24	15.83	27.32	746.61	0.984489
48	48	16.40	28.38	805.34	0.983624

**Table 7.2:** Random Forest metrics based on the combination of parameters

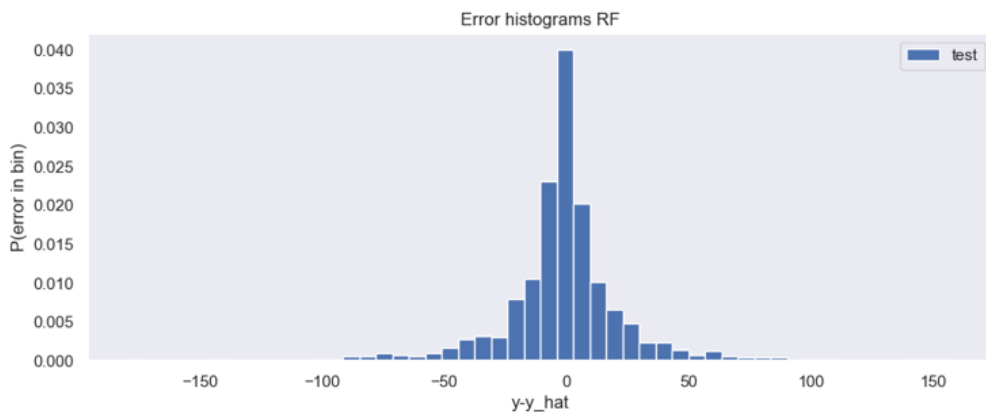
From the table above, we can see that the prediction done with window size equal to 12 hours and horizon equal to 6 hours has the best MAE, which is the first metric that we are going to consider. The second best one is the one with window size equal to 12 hours and horizon equal to 24 hours which has a lower MAE, but has better RMSE, MSE and  $R^2$ .

Looking at these results we find that three different groups can be identified: the first is defined by window size equal to 6 hours, this group identifies the four worst performers by looking at the metrics, then there is the middle group, identified by window size equal to 12 hours and 24 hours, where performance remains stable and the metrics vary slightly, and then there is the last group identified by window size of 48 hours where we can see, again, a deterioration in performance.

### Addition of time features

To see how prediction could be improved and evolved, new features were included for the first two best combination of parameters and for the worst one, that was the one with window size equal to 6 hours and horizon equal to 48 hours. The new features are:

- **day\_before**: represents the data of the day before.
- **week\_before**: represents the data of the week before.
- **hour\_sin**: : represents the sine of the hour of the date we are considering.
- **hour\_cos**: represents represents the cosine of the hour of the date we are considering.
- **month\_sin**: represents the sine of the month of the date we are considering.
- **month\_cos**: represents the cosine of the month of the date we are considering.
- **weekday\_sin**: represents the sine of the day of the week that we are considering.
- **weekday\_cos**: represents the cosine of the day of the week that we are considering.
- **is\_weekend**: : binary value that represents if the day is during weekend (1) or not (0).
- **holiday**: binary value that represents if the day is a holiday or not.



**Figure 7.4:** Errors Random Forest with window size 12 horizon 24

```
Model name: RF
Mean absolute error: 15.78
Mean squared error: 684.85
Root mean squared error: 26.17
      mean      std      MSE      R^2
Test -1.053282  26.148385  684.84745  0.985779
```

Figure 7.5: Evaluation metrics Random Forest with window size 12 horizon 24

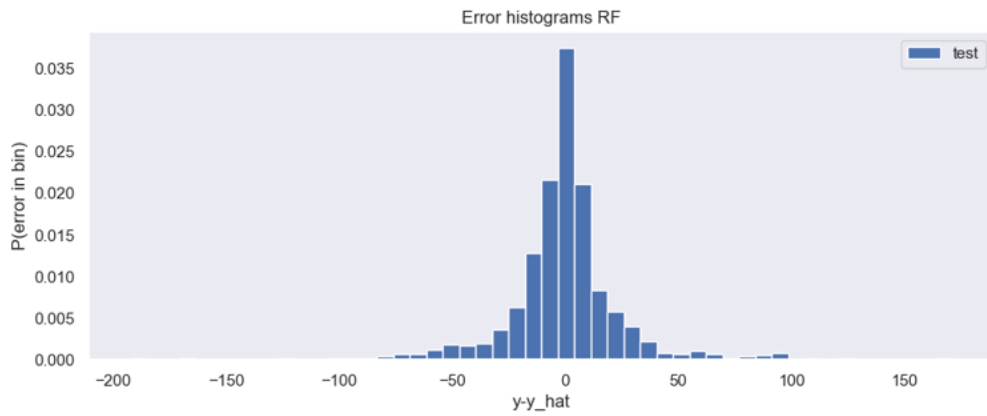
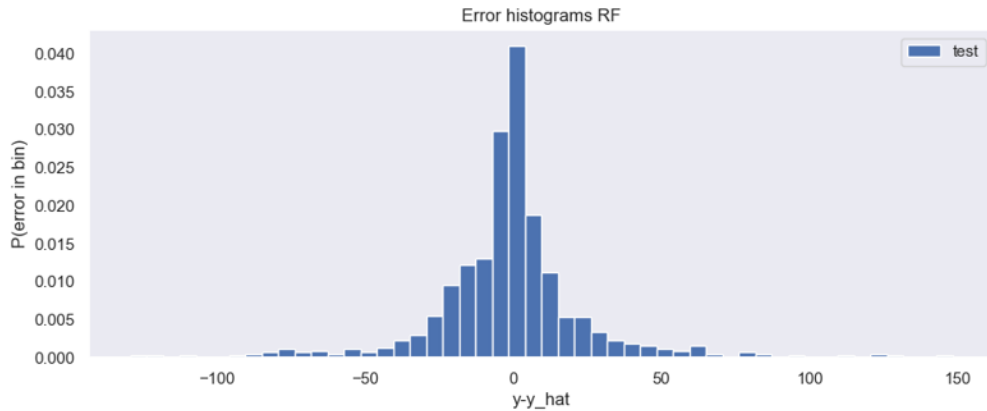


Figure 7.6: Errors Random Forest with window size 12 horizon 6

```
Model name: RF
Mean absolute error: 15.59
Mean squared error: 684.13
Root mean squared error: 26.16
      mean      std      MSE      R^2
Test -0.740344  26.145378  684.128911  0.985837
```

Figure 7.7: Evaluation metrics Random Forest with window size 12 horizon 6



**Figure 7.8:** Errors Random Forest with window size 6 horizon 48

```

Model name: RF
Mean absolute error: 15.75
Mean squared error: 666.09
Root mean squared error: 25.81
      mean      std      MSE      R^2
Test -1.092034  25.785682  666.093934  0.98646
    
```

**Figure 7.9:** Evaluation metrics Random Forest with window size 6 horizon 48

The graphs above show that the evaluation metrics for the two best parameter combinations are slightly worsened by the addition of these temporal features, with the exception of the MAE of the model with a window size of 12 and a horizon of 6, which stays the same. If we concentrate on the parameter combination with the worst performance, we can see that all of the evaluation metrics significantly improve.

### Addition of weather features

Then, weather features were added, where the only feature was "RAINED", that is a boolean value that says if during that day it rained or not.

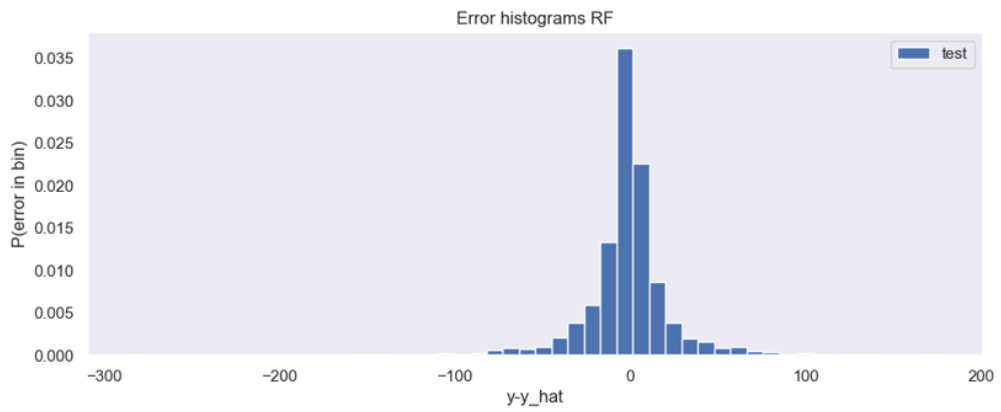


Figure 7.10: Errors Random Forest with window size 12 horizon 24

```
Model name: RF
Mean absolute error: 15.59
Mean squared error: 752.60
Root mean squared error: 27.43
      mean      std      MSE      R^2
Test -1.709497  27.380186  752.596971  0.984407
```

Figure 7.11: Evaluation metrics Random Forest with window size 12 horizon 24

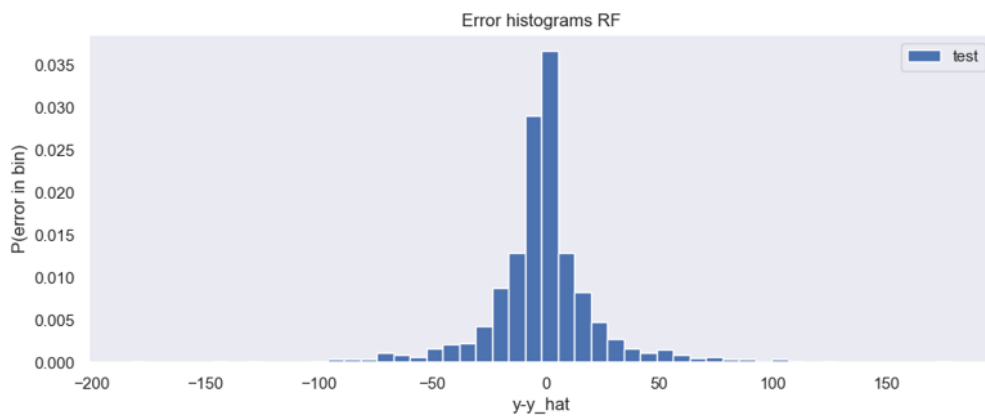
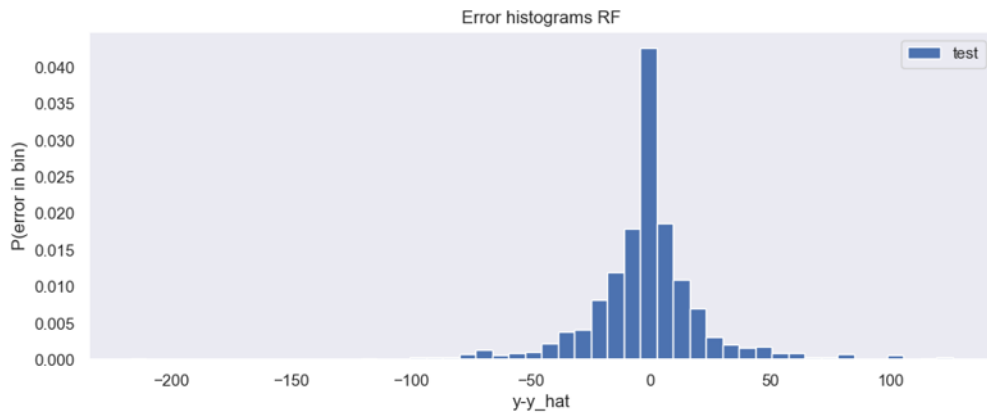


Figure 7.12: Errors Random Forest with window size 12 horizon 6

```

Model name: RF
Mean absolute error: 15.71
Mean squared error: 690.68
Root mean squared error: 26.28
      mean      std      MSE      R^2
Test -1.497736  26.238091  690.680634  0.985737
    
```

**Figure 7.13:** Evaluation metrics Random Forest with window size 12 horizon 6



**Figure 7.14:** Errors Random Forest with window size 6 horizon 48

```

Model name: RF
Mean absolute error: 15.85
Mean squared error: 680.67
Root mean squared error: 26.09
      mean      std      MSE      R^2
Test -2.028596  26.010671  680.670187  0.986222
    
```

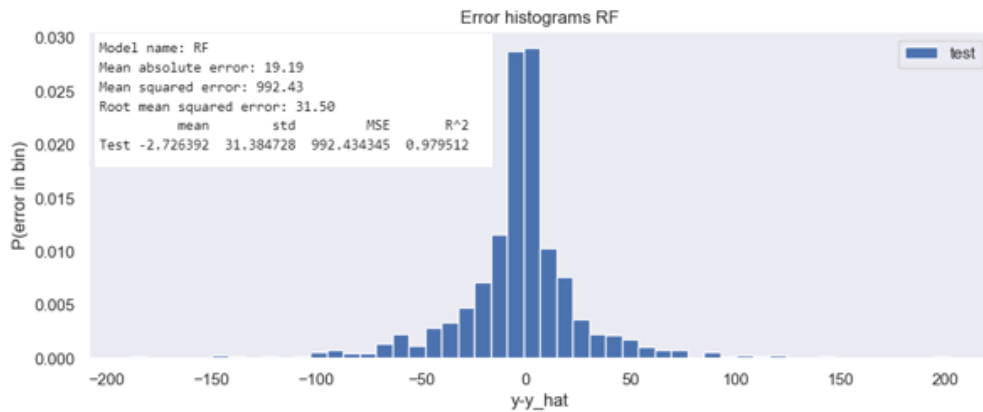
**Figure 7.15:** Evaluation metrics Random Forest with window size 6 horizon 48

The graphs above demonstrate that, with the exception of the MAE of the model with window size equal to 12 and horizon equal to 24, the addition of these weather elements slightly degrades the evaluation metrics for the two best parameter combinations. If we focus on the parameter combination that performs the worst, we can see that all of the evaluation metrics improve compared to when the window size is the only feature, but they slightly degrade compared to when the window size and the temporal feature are features.

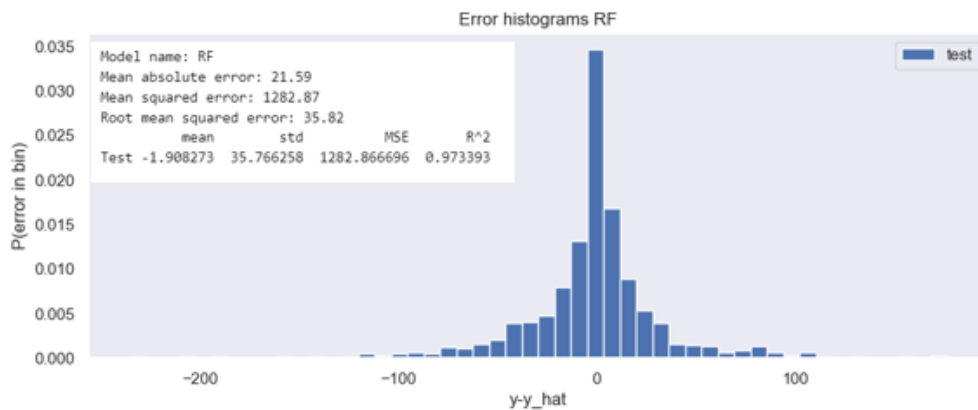


### 7.1.4 Third experimental session

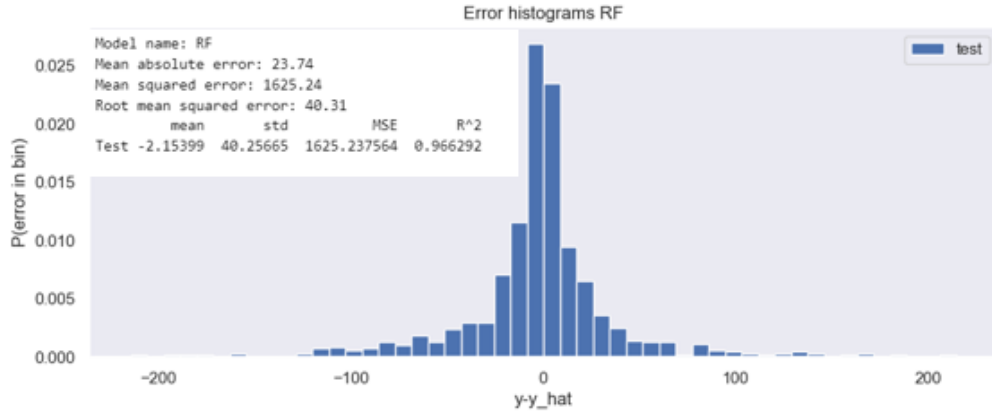
The best model identified in the previous phase was used in this session, and the "lead-time" parameter was modified to examine how the prediction performs when the most recent data are not available.



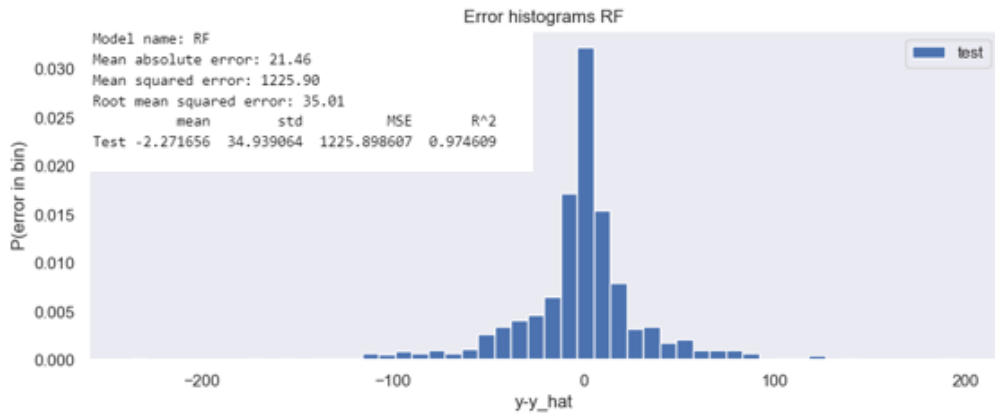
**Figure 7.16:** Error histograms and evaluation metrics for prediction with lead time equal to 6



**Figure 7.17:** Error histograms and evaluation metrics for prediction with lead time equal to 12



**Figure 7.18:** Error histograms and evaluation metrics for prediction with lead time equal to 24



**Figure 7.19:** Error histograms and evaluation metrics for prediction with lead time equal to 48

The graphs and evaluation metrics show that, when compared to the situation where the lead time is equal to 1 (7.2), the prediction consistently performs poorly. Compared to the situations where the lead time is equal to 12 and 24, it is clear that the evaluation metrics are superior when the lead time is equal to 48.

### 7.1.5 Machine learning regression models comparison

These tests were carried out using an ideal configuration of the algorithms, which was achieved by fine-tuning the hyperparameters using a grid search.

Model name: RF  
Mean absolute error: 19.80  
Mean squared error: 1227.92  
Root mean squared error: 35.04  
Test -1.302974 35.017448 1227.919387 0.974457

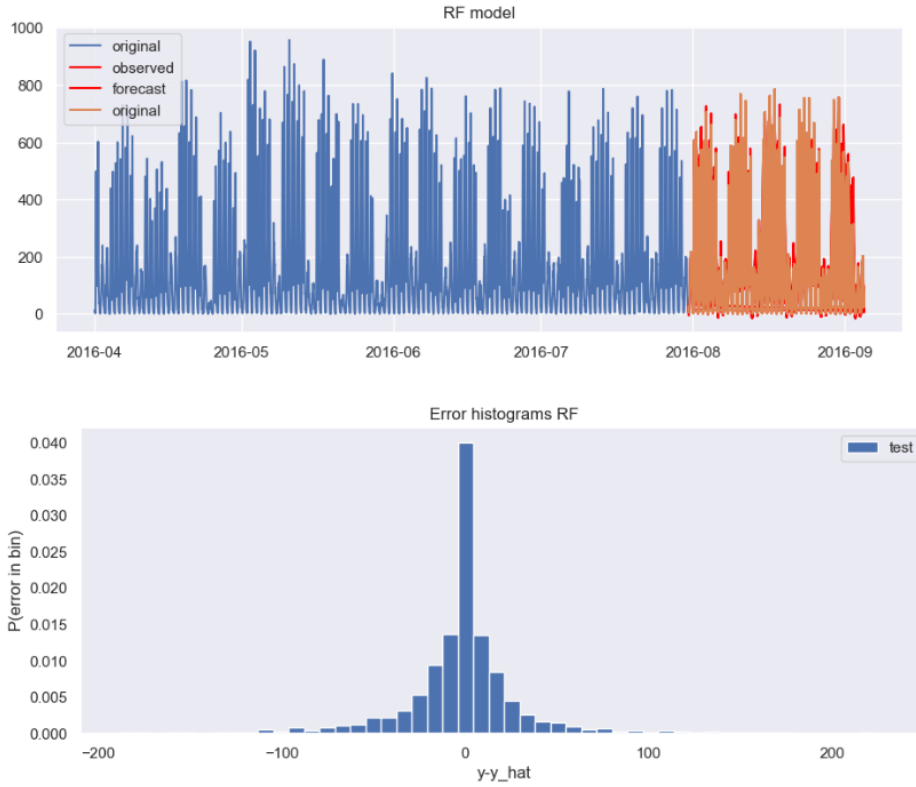
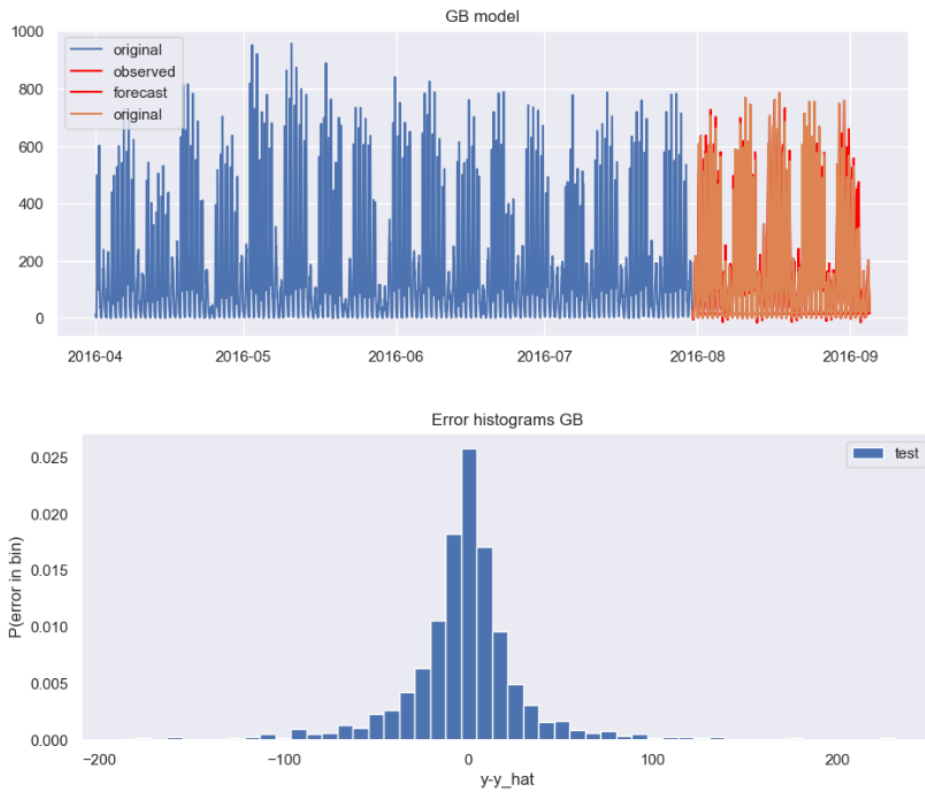


Figure 7.20: Random forest with base configuration

## Experimental analysis

Model name: GB  
Mean absolute error: 21.83  
Mean squared error: 1281.65  
Root mean squared error: 35.80  
Test -1.503122 35.768632 1281.654414 0.973364

	mean	std	MSE	R <sup>2</sup>
Test	-1.503122	35.768632	1281.654414	0.973364



**Figure 7.21:** Gradient Boosting with base configuration

Model name: Lasso  
Mean absolute error: 23.49  
Mean squared error: 1442.69  
Root mean squared error: 37.98  
Test -1.619554 37.948247 1442.692375 0.97001

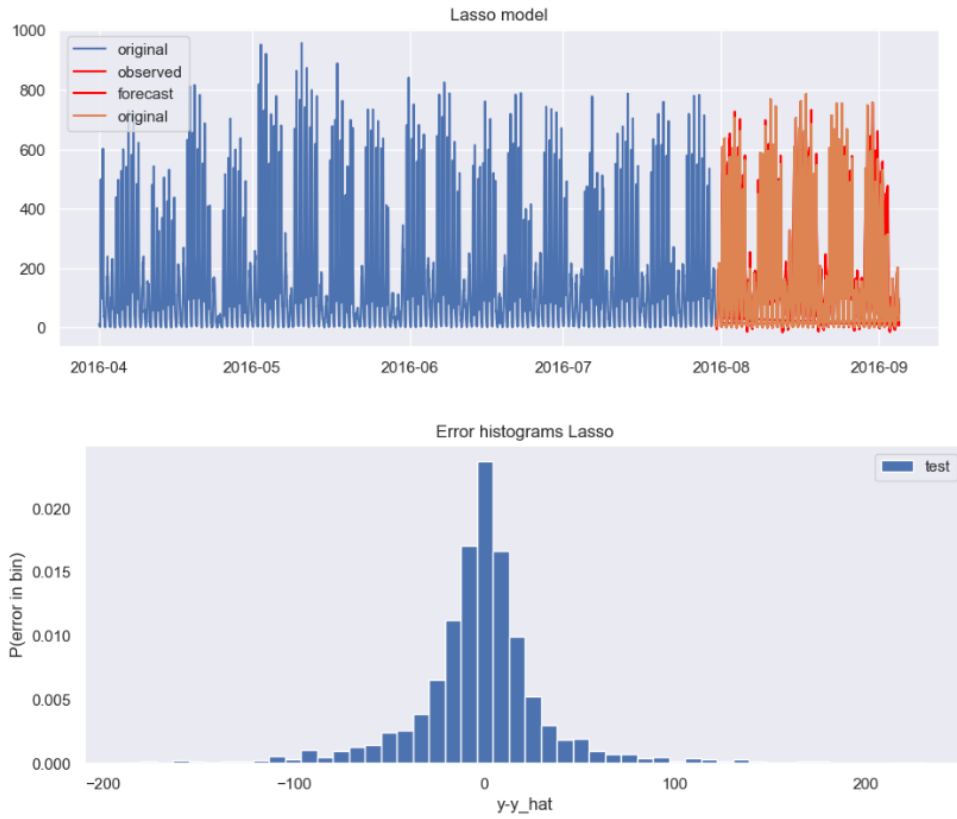


Figure 7.22: Lasso with base configuration

Model name: LR  
Mean absolute error: 24.36  
Mean squared error: 1524.02  
Root mean squared error: 39.04  
Test mean std MSE R^2  
Test -1.517569 39.009242 1524.023972 0.968304

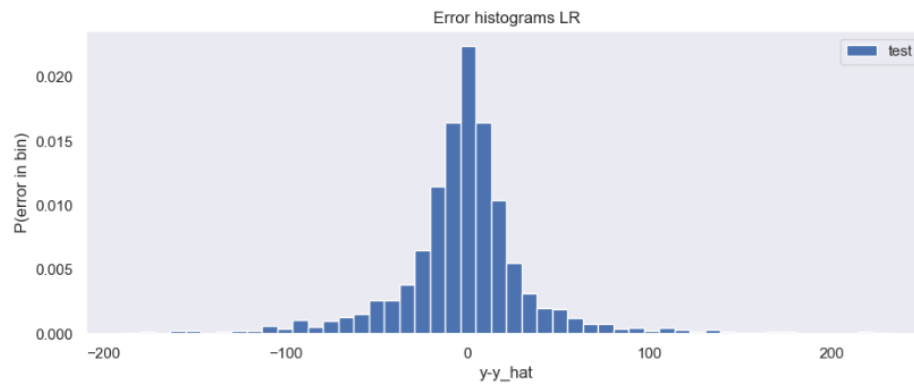
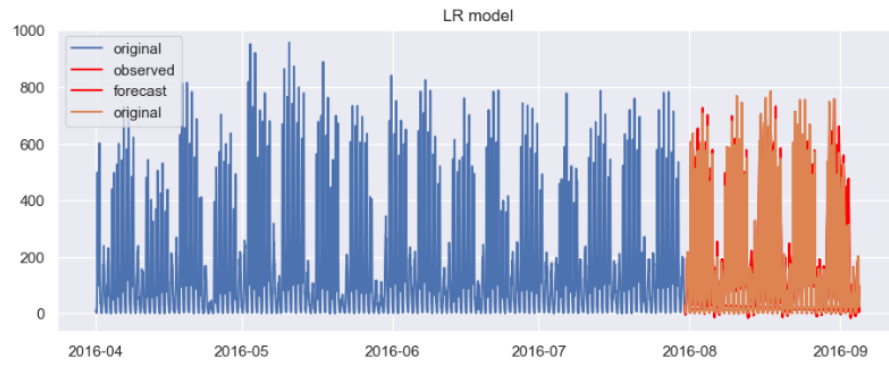


Figure 7.23: Linear Regression with base configuration

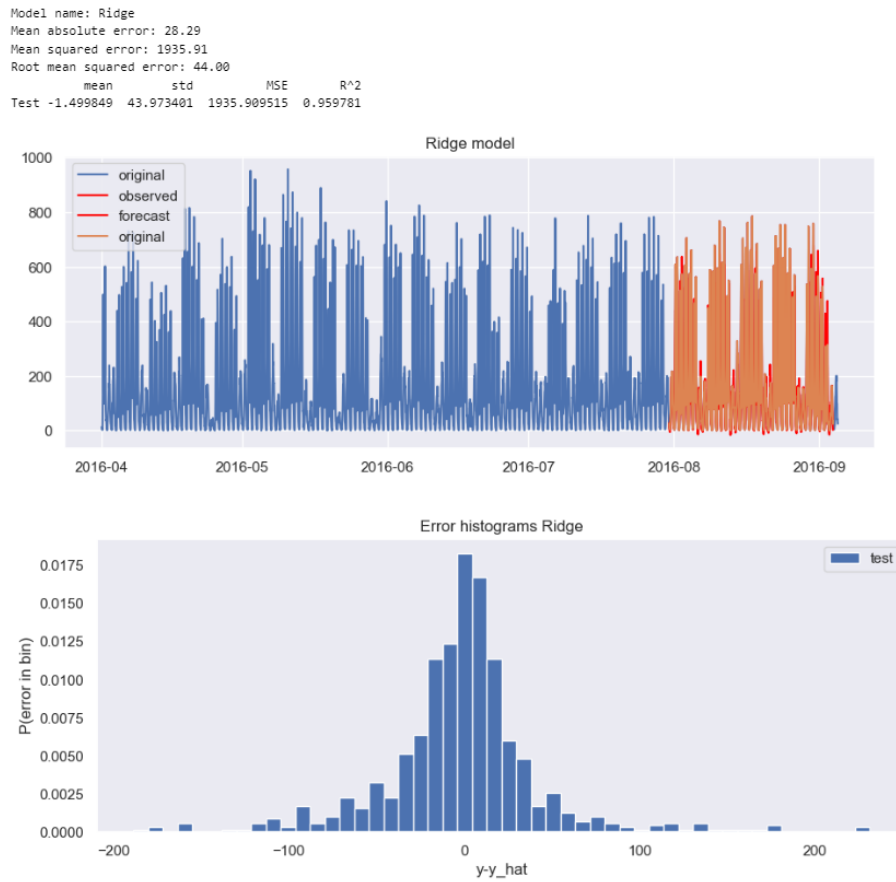


Figure 7.24: Ridge with base configuration

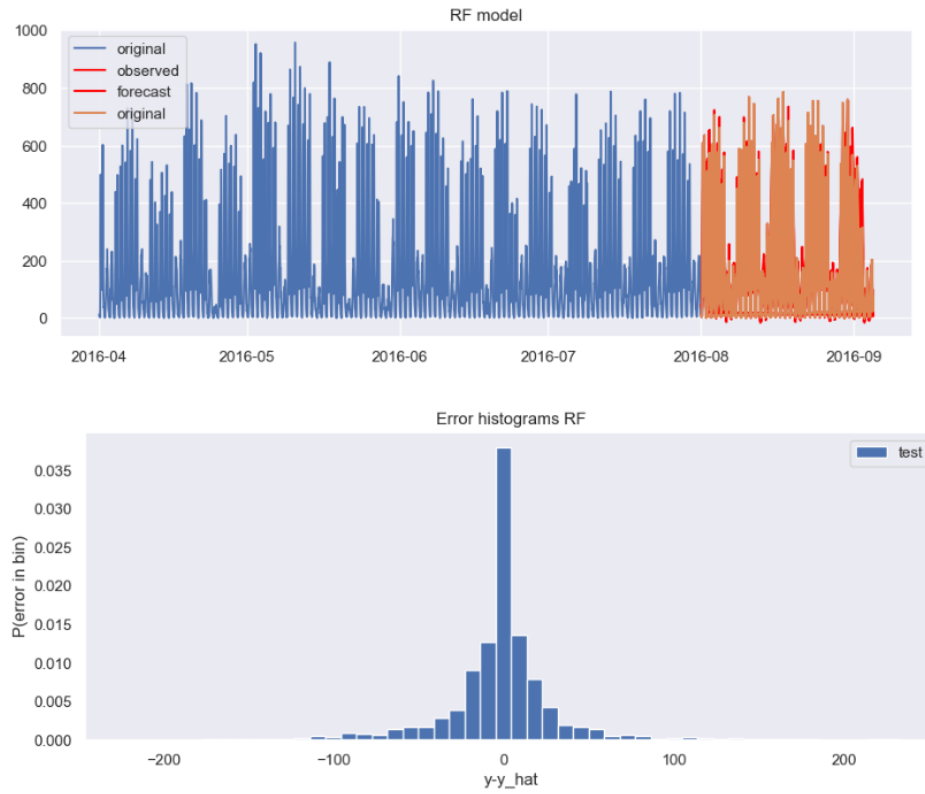
The Random Forest model is the best one, as can be shown from the evaluation metrics.

Let's investigate if the Random Forest model still performs the best when the value of the horizon is changed.

## Experimental analysis

Model name: RF  
Mean absolute error: 19.98  
Mean squared error: 1253.10  
Root mean squared error: 35.40  
Test

	mean	std	MSE	R <sup>2</sup>
Test	-1.659052	35.360289	1253.102506	0.973807



**Figure 7.25:** Random Forest with horizon 48



Model name: GB  
Mean absolute error: 22.06  
Mean squared error: 1305.07  
Root mean squared error: 36.13  
Test    mean    std    MSE    R<sup>2</sup>  
Test -1.758955 36.082926 1305.071493 0.973016

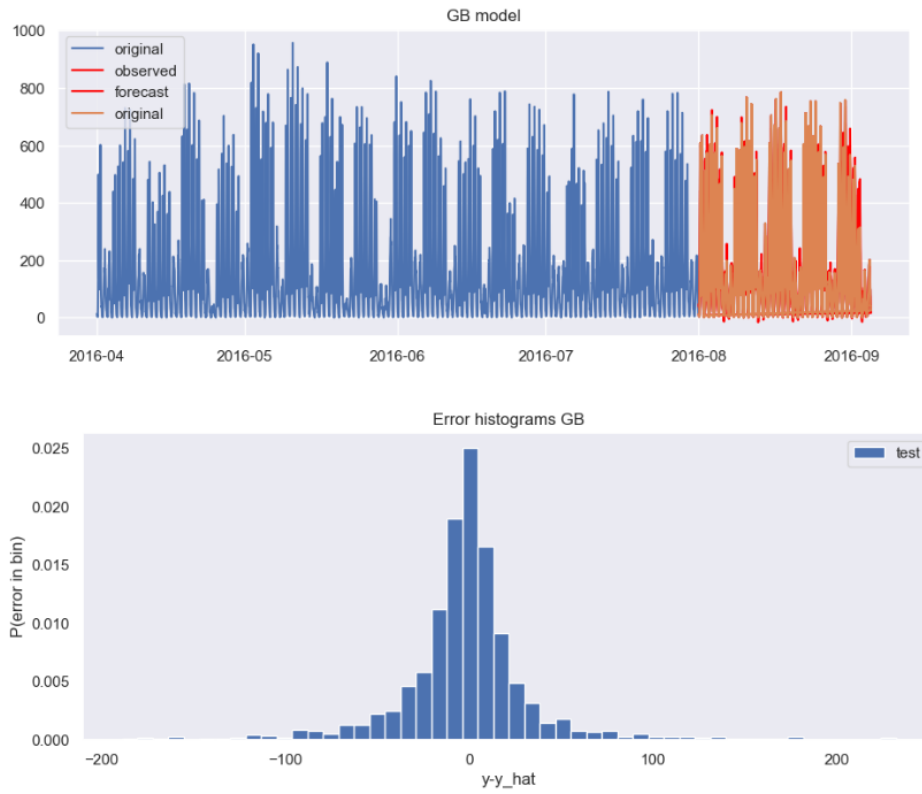
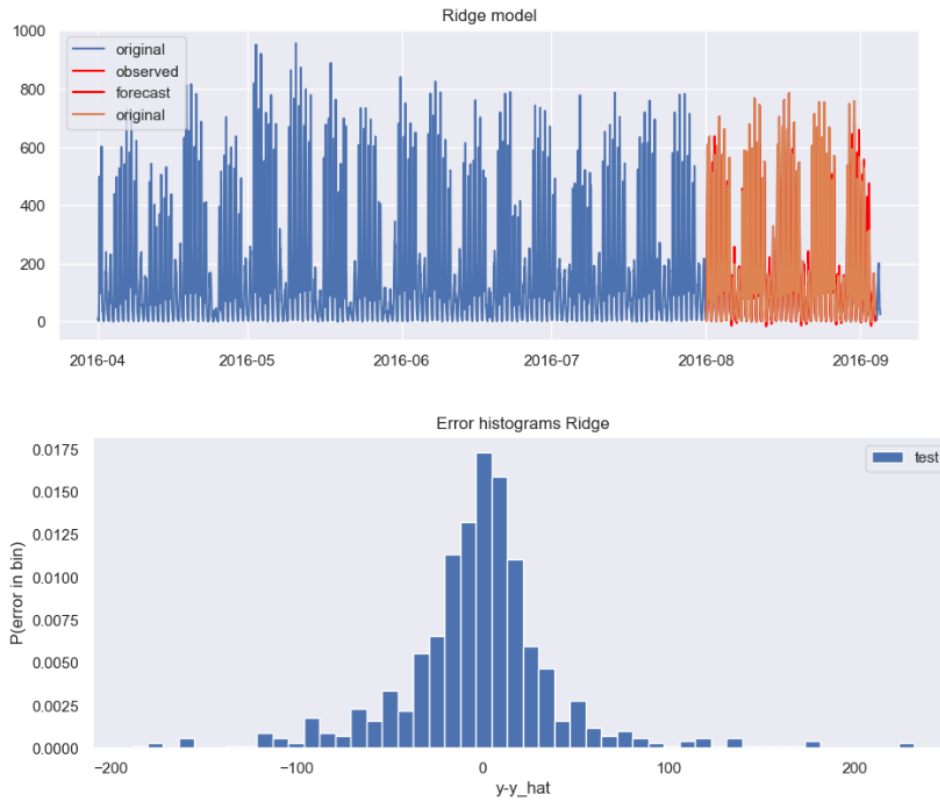


Figure 7.26: Gradient Boosting with horizon 48

## Experimental analysis

Model name: Ridge  
Mean absolute error: 28.88  
Mean squared error: 1995.50  
Root mean squared error: 44.67

	mean	std	MSE	R <sup>2</sup>
Test	-1.834958	44.633344	1995.502494	0.959431



**Figure 7.27:** Ridge with horizon 48

Model name: Lasso  
Mean absolute error: 23.57  
Mean squared error: 1455.07  
Root mean squared error: 38.15  
Test

	mean	std	MSE	R <sup>2</sup>
Test	-1.892093	38.098435	1455.070775	0.969738

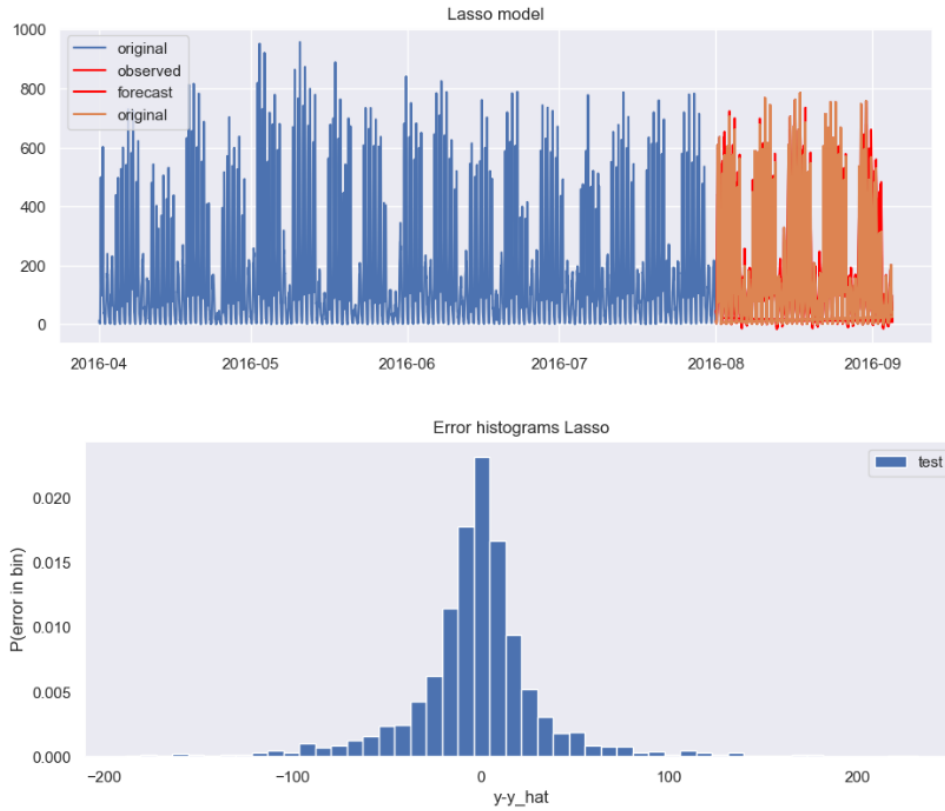


Figure 7.28: Lasso with horizon 48

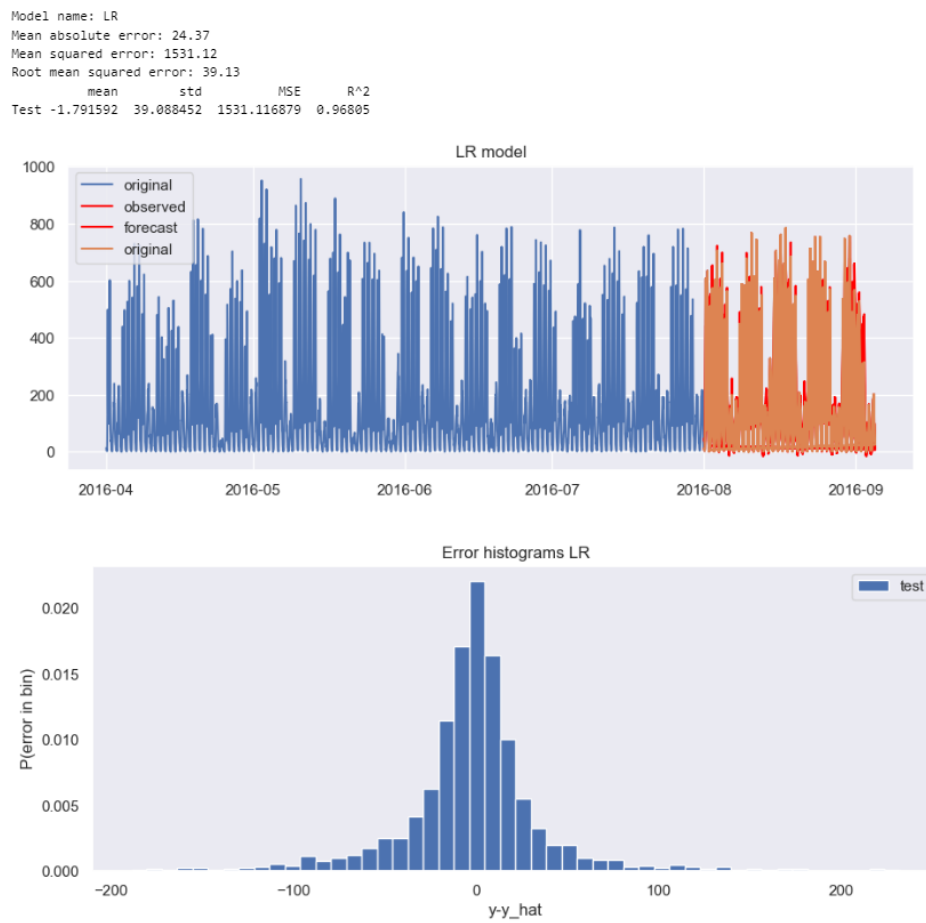


Figure 7.29: Linear Regression with horizon 48

As can be seen, the Random Forest model continues to be the best performing.

## 7.2 San Francisco dataset

### 7.2.1 ARIMA



Figure 7.30: ARIMA forecasting for San Francisco dataset

MAE	MSE	$R^2$	MAPE
0.987	2.125	0.725	0.071

Table 7.3: ARIMA evaluation metrics San Francisco

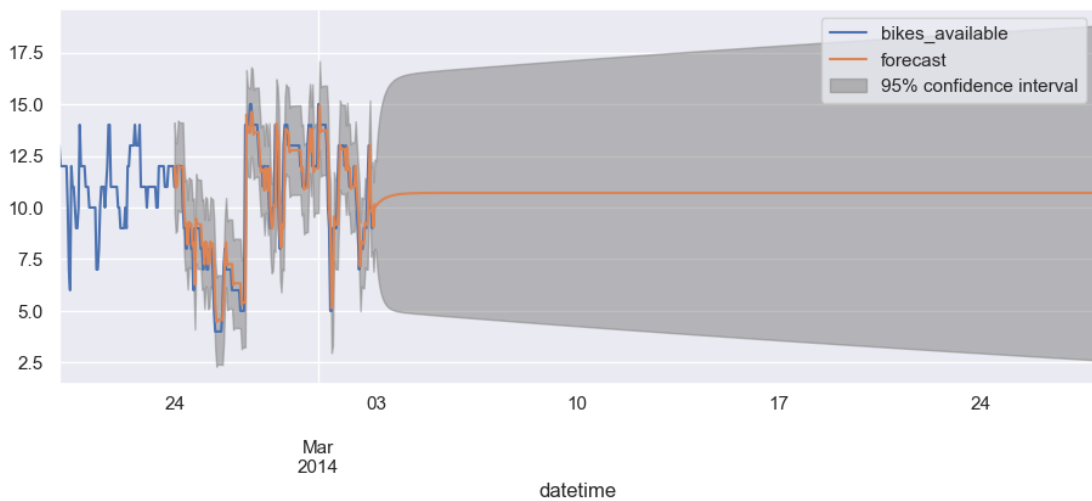


Figure 7.31: ARIMA forecasting for San Francisco dataset

According to the plots and the table above, even though the forecast curve resembles the curve of actual values, this model does not perform very well: the  $R^2$  score is a respectable value, the MSE value is quite low and that means that the values are dispersed closely to the mean, so the errors are smaller and the estimation is better, the MAPE is very low and that means that the forecasted values are quite similar to the actual ones. Since our values range from approximately 4 to approximately 15, the MAE is quite good too. But, all things considered, this model does not perform particularly well: we can see that it predicts good only for short term data but it is not a good model for long-term predictions as we can see in picture 7.31.

## **7.2.2 First experimental session**

In order to determine which regressor could most accurately predict the bike occupancy in the chosen station, the first round of experiments involved applying each regressor to the dataset in question. To do this, the horizon parameter was changed, and the other parameters were configured using a base configuration: lags (window size) equal to 6 (hours), lead\_time equal to 1, size equal to 24\*120, size\_test equal to 24.



**Figure 7.32:** Random Forest for San Francisco dataset, prediction made using base configuration

Comparing each output obtained from the application of the regressors, due to the evaluation metrics considered (RMSE, MSE, MAE,  $R^2$ ), the Random Forest was found to be the best regressor among all, having each metric better than all other regressors considered.

### 7.2.3 Second experimental session

Since Random Forest turned out to be the best regression model, in the second round of experiments we focused on finding the best parameters for the sliding window technique using this regression model. For this round of experiments we decided to use as features only the instants that make up the sliding window, for example if the window size is equal to 6 hours, the features taken into account will

be the current hour plus the previous 5 hours taken individually.

Window size	horizon	MAE	RMSE	MSE	$R^2$
6	6	0.76	1.33	1.78	0.986314
6	12	0.78	1.34	1.79	0.986218
6	24	0.79	1.36	1.84	0.985771
6	48	0.83	1.38	1.90	0.985338
<b>12</b>	<b>6</b>	<b>0.70</b>	<b>1.25</b>	<b>1.56</b>	<b>0.987907</b>
12	12	0.72	1.29	1.67	0.987062
12	24	0.73	1.33	1.76	0.986364
12	48	0.76	1.33	1.76	0.986283
<b>24</b>	<b>6</b>	<b>0.69</b>	<b>1.24</b>	<b>1.53</b>	<b>0.988009</b>
24	12	0.71	1.28	1.65	0.987058
24	24	0.76	1.29	1.67	0.986866
24	48	0.77	1.32	1.75	0.986228
48	6	0.74	1.30	1.70	0.986762
48	12	0.77	1.31	1.71	0.986618
48	24	0.81	1.34	1.79	0.985995
48	48	0.84	1.37	1.86	0.985322

**Table 7.4:** Random Forest metrics based on the combination of parameters

From the table above, we can see that the prediction done with window size equal to 24 hours and horizon equal to 6 hours has the best parameters. The second best one is the one with window size equal to 12 hours and horizon equal to 6 hours which has slightly worse parameters.

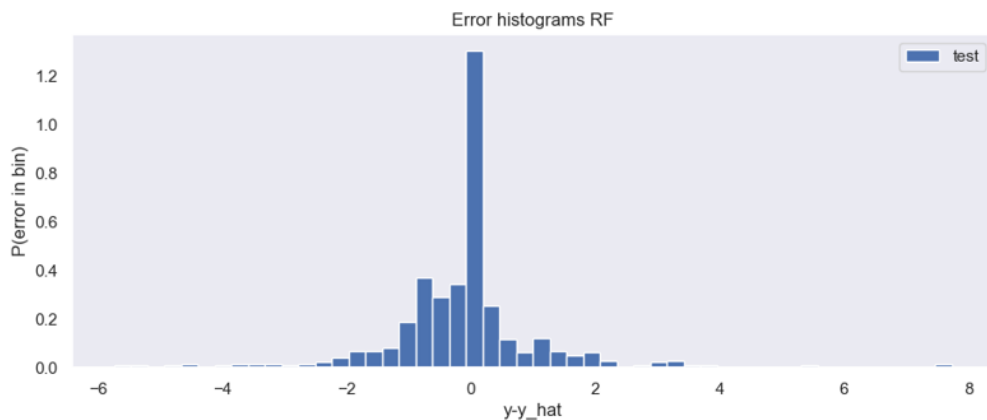
Looking at these results we find that three different groups can be identified: the first is defined by window size equal to 6 hours, this group identifies the four worst performers by looking at the metrics, then there is the middle group, identified by window size equal to 12 hours and 24 hours, where performance remains stable and the metrics vary slightly, and then there is the last group identified by window size of 48 hours where we can see, again, a deterioration in performance.



### Addition of time features

To see how prediction could be improved and evolved, new features were included for the first two best combination of parameters and for the two worse, that were the one with window size equal to 6 hours and horizon equal to 48 hours and the one with window size equal to 48 hours and horizon equal to 48 hours. The new features are:

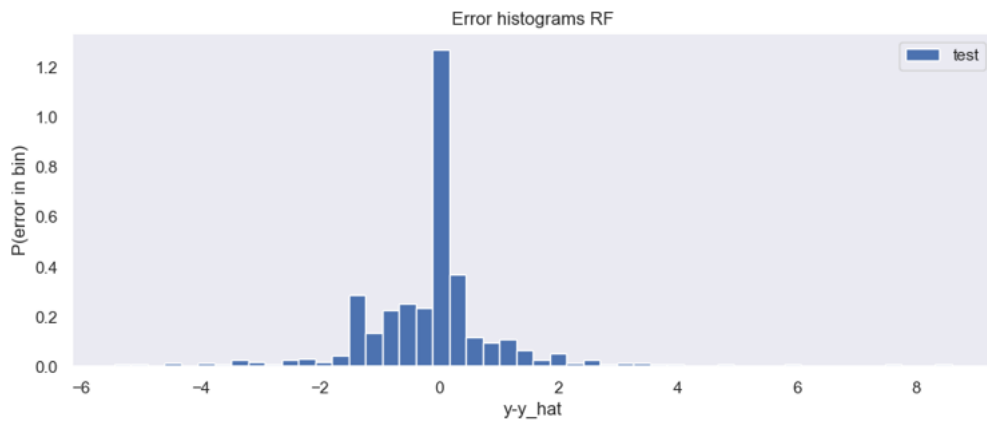
- **day\_before**: represents the data of the day before.
- **week\_before**: represents the data of the week before.
- **hour\_sin**: : represents the sine of the hour of the date we are considering.
- **hour\_cos**: represents represents the cosine of the hour of the date we are considering.
- **month\_sin**: represents the sine of the month of the date we are considering.
- **month\_cos**: represents the cosine of the month of the date we are considering.
- **weekday\_sin**: represents the sine of the day of the week that we are considering.
- **weekday\_cos**: represents the cosine of the day of the week that we are considering.
- **is\_weekend**: : binary value that represents if the day is during weekend (1) or not (0).
- **holiday**: binary value that represents if the day is a holiday or not.



**Figure 7.33:** Random Forest error histograms with window size 24 horizon 6

MAE	MSE	RMSE	$R^2$
0.70	1.47	1.21	0.988651

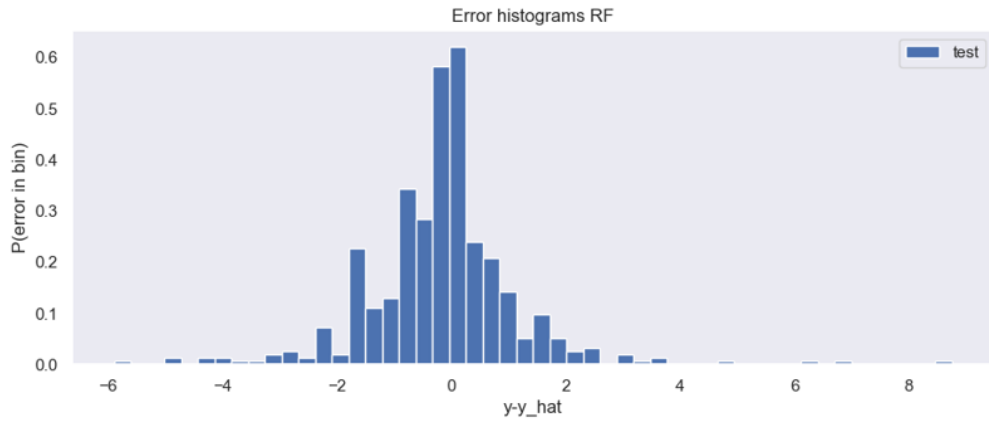
**Table 7.5:** Evaluation metrics window size 24 horizon 6



**Figure 7.34:** Random Forest error histograms with window size 12 horizon 6

MAE	MSE	RMSE	$R^2$
0.74	1.56	1.25	0.987917

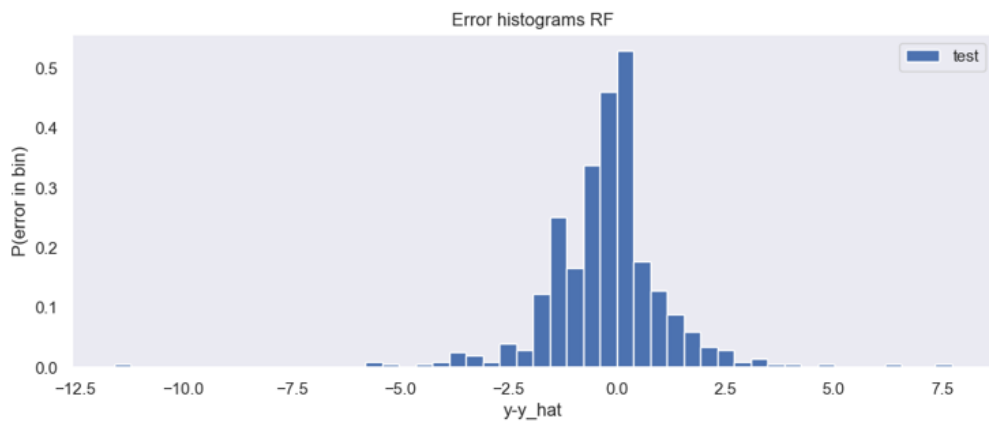
**Table 7.6:** Evaluation metrics window size 12 horizon 6



**Figure 7.35:** Random Forest error histograms with window size 6 horizon 48

MAE	MSE	RMSE	$R^2$
0.86	1.75	1.32	0.986416

**Table 7.7:** Evaluation metrics window size 6 horizon 48



**Figure 7.36:** Random Forest error histograms with window size 48 horizon 48

MAE	MSE	RMSE	$R^2$
0.92	2.06	1.44	0.984189

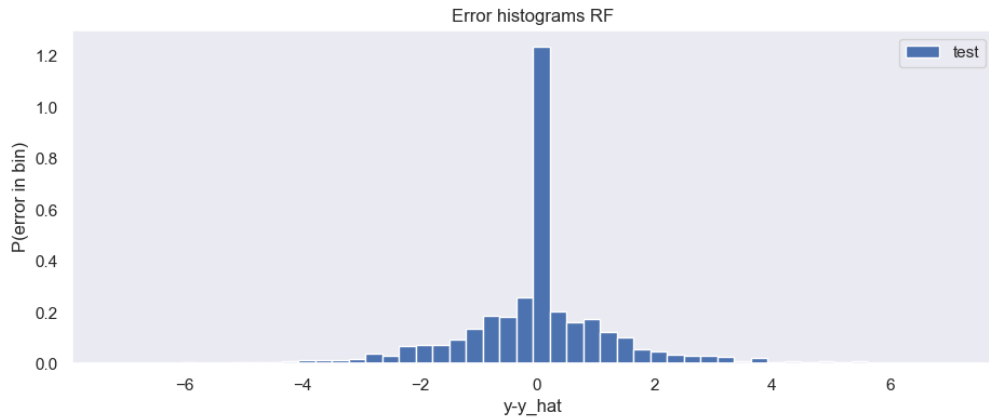
**Table 7.8:** Evaluation metrics window size 48 horizon 48

The graphs and tables above show that the evaluation metrics for the best parameters combination improve except for the MAE that worsen slightly. For the second best combination the evaluation metrics worsen if we concentrate on MAE, stays the same if we concentrate on RMSE and MSE and improves if we concentrate on  $R^2$ . For the combination window size equal to 6 and horizon equal to 48 MAE worsens, instead the other metrics improve. For the combination window size equal to 48 and horizon equal to 48 every metric worsens.

### Addition of weather features

Then, weather features were added, where the features that were chosen to take into account are the following:

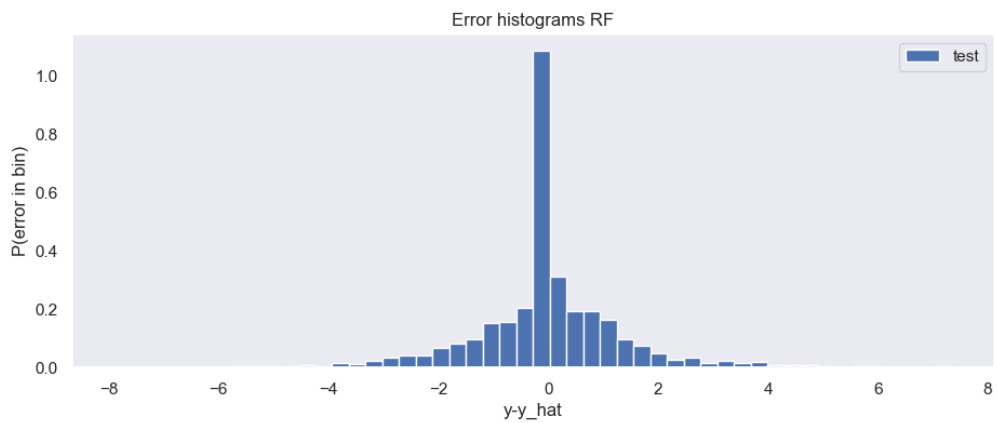
- **precipitation\_inches:** represents the amount of inches of rain.
- **cloud\_cover:** is a measure of how much the sky is covered by clouds. It is measured in oktas, if the value is equal to 8 the sky is completely overcast.
- **Fog:** Boolean value that indicates whether there is fog or not.
- **No events:** Boolean value that indicates whether there is a meteorological event or not.
- **Rain:** Boolean value that indicates if it rained or not.



**Figure 7.37:** Random Forest error histograms with window size 24 horizon 6

MAE	MSE	RMSE	$R^2$
0.86	1.94	1.39	0.986887

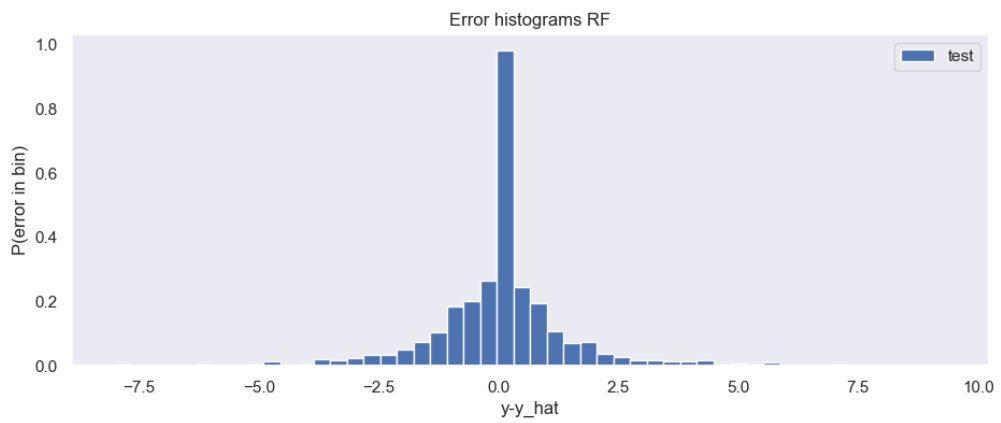
**Table 7.9:** Evaluation metrics window size 24 horizon 6



**Figure 7.38:** Random Forest error histograms with window size 12 horizon 6

MAE	MSE	RMSE	$R^2$
0.85	1.91	1.38	0.987132

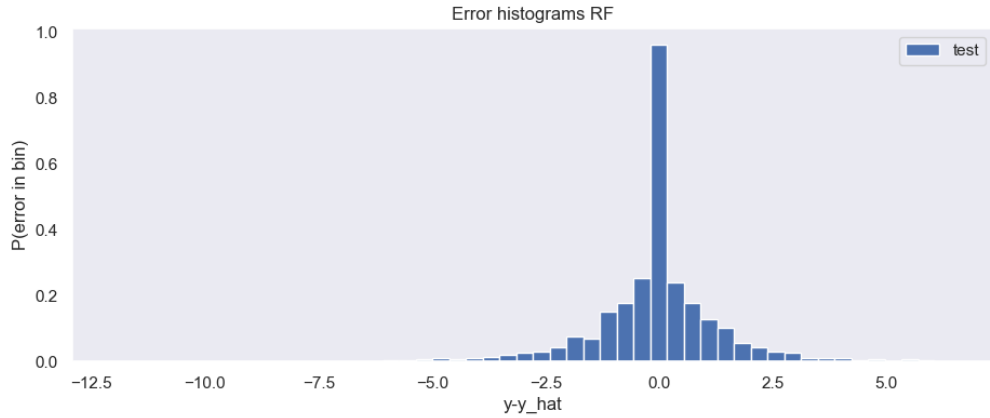
**Table 7.10:** Evaluation metrics window size 12 horizon 6



**Figure 7.39:** Random Forest error histograms with window size 6 horizon 48

MAE	MSE	RMSE	$R^2$
0.87	2.08	1.44	0.985926

**Table 7.11:** Evaluation metrics window size 6 horizon 48



**Figure 7.40:** Random Forest error histograms with window size 48 horizon 48

MAE	MSE	RMSE	$R^2$
0.85	1.93	1.39	0.986932

**Table 7.12:** Evaluation metrics window size 48 horizon 48

The graphs and tables above demonstrate that the addition of these weather elements slightly degrades the evaluation metrics for the two best parameter combinations, but looking at the combination with window size 6 and horizon 48 and the combination of window size 48 and horizon 48 it can be seen that their  $R^2$  score improves.

### 7.2.4 Third experimental session

The best model identified in the previous phase was used in this session, and the "lead-time" parameter was modified to examine how the prediction performs when the most recent data are not available.

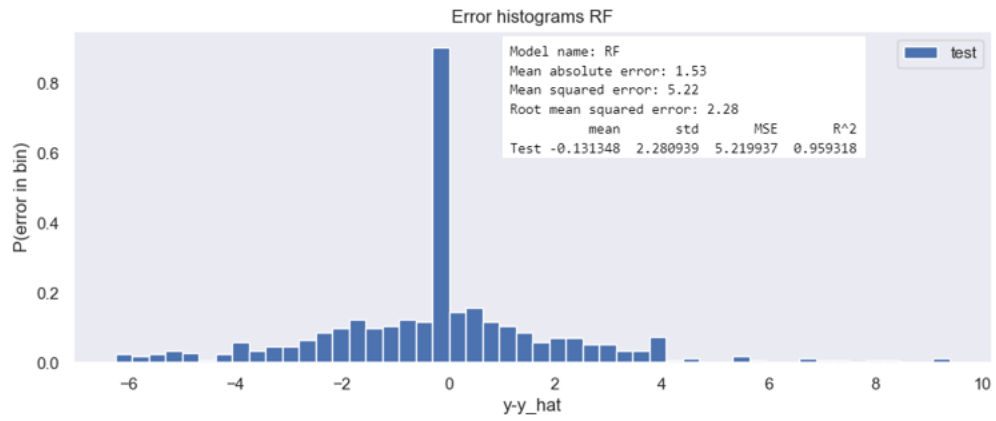


Figure 7.41: Error histograms for prediction with lead time equal to 6, plus evaluation metrics

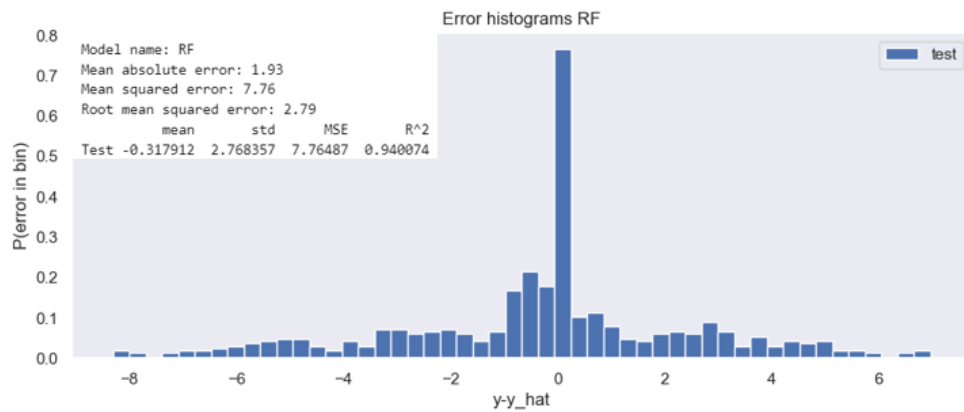
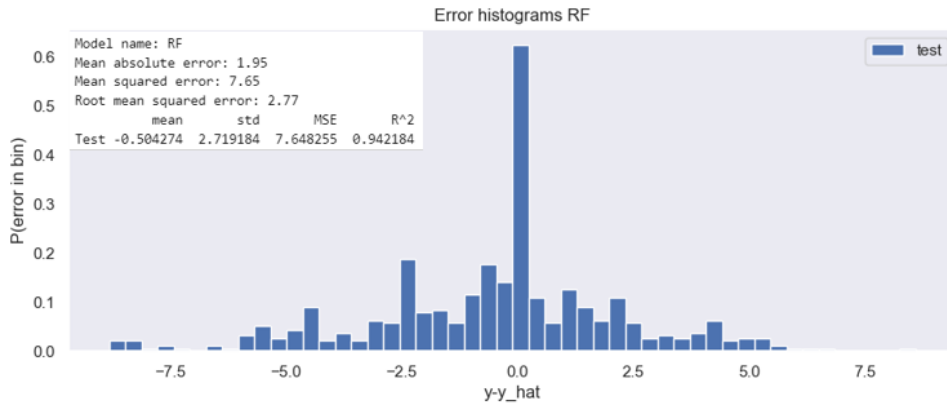
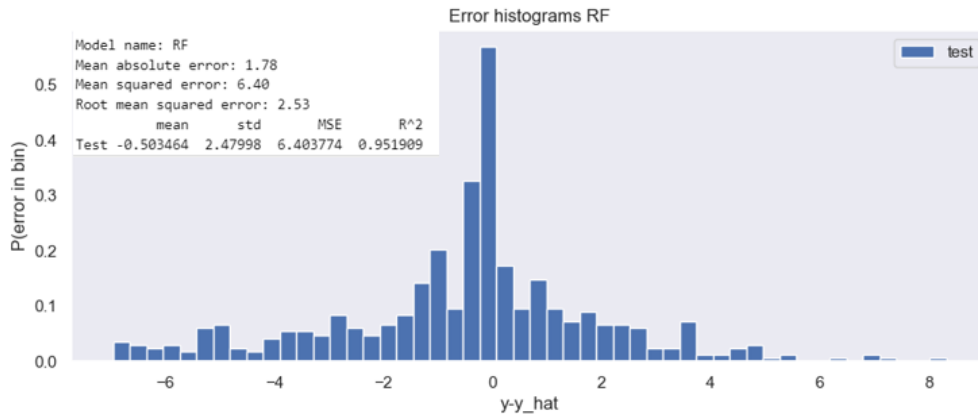


Figure 7.42: Error histograms for prediction with lead time equal to 12, plus evaluation metrics





**Figure 7.43:** Error histograms for prediction with lead time equal to 24, plus evaluation metrics



**Figure 7.44:** Error histograms for prediction with lead time equal to 48, plus evaluation metrics

The graphs and evaluation metrics show that, when compared to the situation where the lead time is equal to 1 (7.4), the prediction consistently performs poorly. Compared to the situations where the lead time is equal to 12, it is clear that the evaluation metrics are superior when the lead time is equal to 48. When lead time is equal to 24 there is a slightly improvement for the MSE, RMSE and  $R^2$  compared to lead time equal to 12.

### 7.2.5 Machine learning regression models comparison

These tests were carried out using an ideal configuration of the algorithms, which was achieved by fine-tuning the hyperparameters using a grid search.

```
Training window:  
...  
Mean squared error: 1.01  
Root mean squared error: 1.00  
      mean      std      MSE      R^2  
Test -0.059947  1.003155  1.009914  0.991998
```



Figure 7.45: Random Forest with base configuration

```
Training window:  
...  
Mean squared error: 1.16  
Root mean squared error: 1.08  
      mean      std      MSE      R^2  
Test -0.059151  1.077324  1.164127  0.990809
```

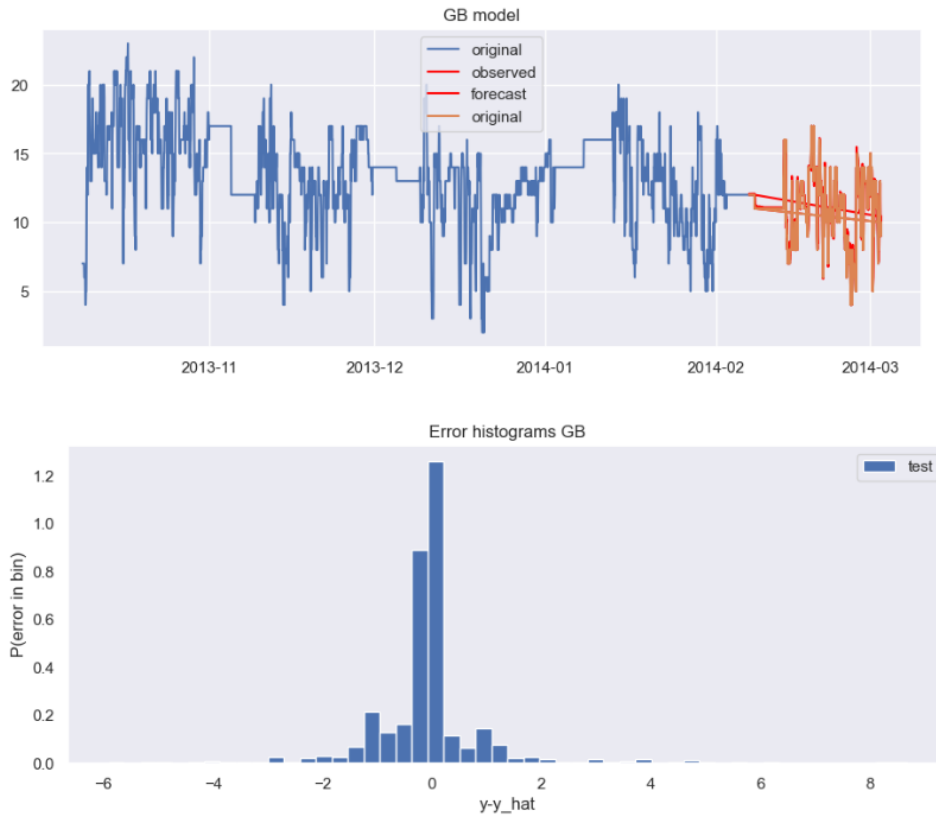


Figure 7.46: Gradient Boosting with base configuration

```
Training window:  
...  
Mean squared error: 1.15  
Root mean squared error: 1.07  
      mean      std      MSE      R^2  
Test -0.07445  1.067618  1.145352  0.991038
```



Figure 7.47: Ridge with base configuration

```
Training window:  
...  
Mean squared error: 1.17  
Root mean squared error: 1.08  
      mean      std      MSE      R^2  
Test -0.064786  1.081145  1.173071  0.990722
```

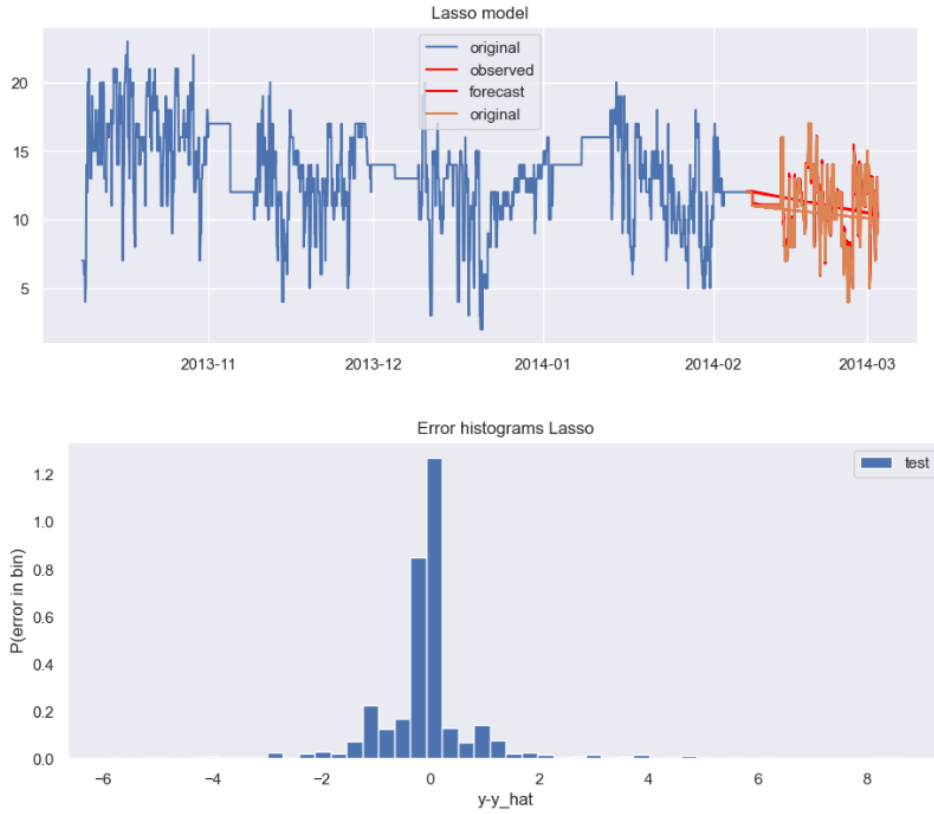


Figure 7.48: Lasso with base configuration



**Figure 7.49:** Linear Regression with base configuration

The Random Forest model is the best one, as can be shown from the evaluation metrics.

Let's investigate if the Random Forest model still performs the best when the value of the horizon is changed.

```
Training window:  
...  
Mean squared error: 1.08  
Root mean squared error: 1.04  
      mean      std      MSE      R^2  
Test -0.056651  1.038213  1.081095  0.991429
```

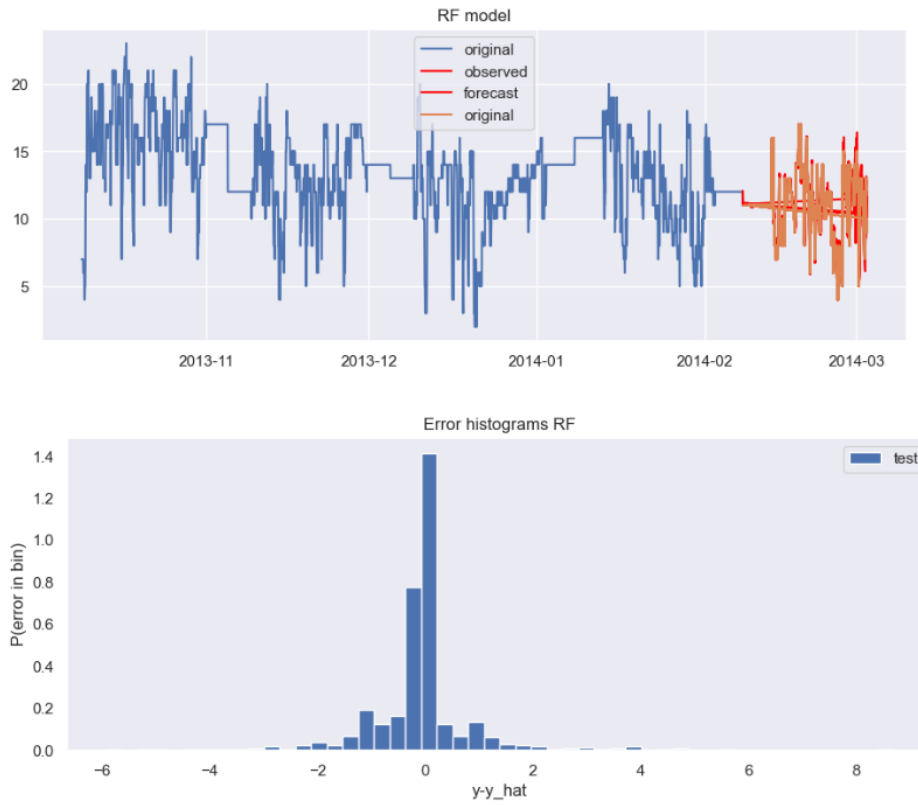


Figure 7.50: Random Forest with horizon 48

```
Training window:  
...  
Mean squared error: 1.22  
Root mean squared error: 1.10  
      mean      std      MSE      R^2  
Test -0.058897  1.101797  1.217425  0.990366
```



Figure 7.51: Gradient Boosting with horizon 48



```
Training window:  
...  
Mean squared error: 1.20  
Root mean squared error: 1.09  
      mean      std      MSE      R^2  
Test -0.076932  1.091636  1.197587  0.990573
```



Figure 7.52: Ridge with horizon 48

```
Training window:  
...  
Mean squared error: 1.23  
Root mean squared error: 1.11  
Test  
mean    std    MSE    R^2  
-0.063237 1.1053 1.225687 0.990294
```

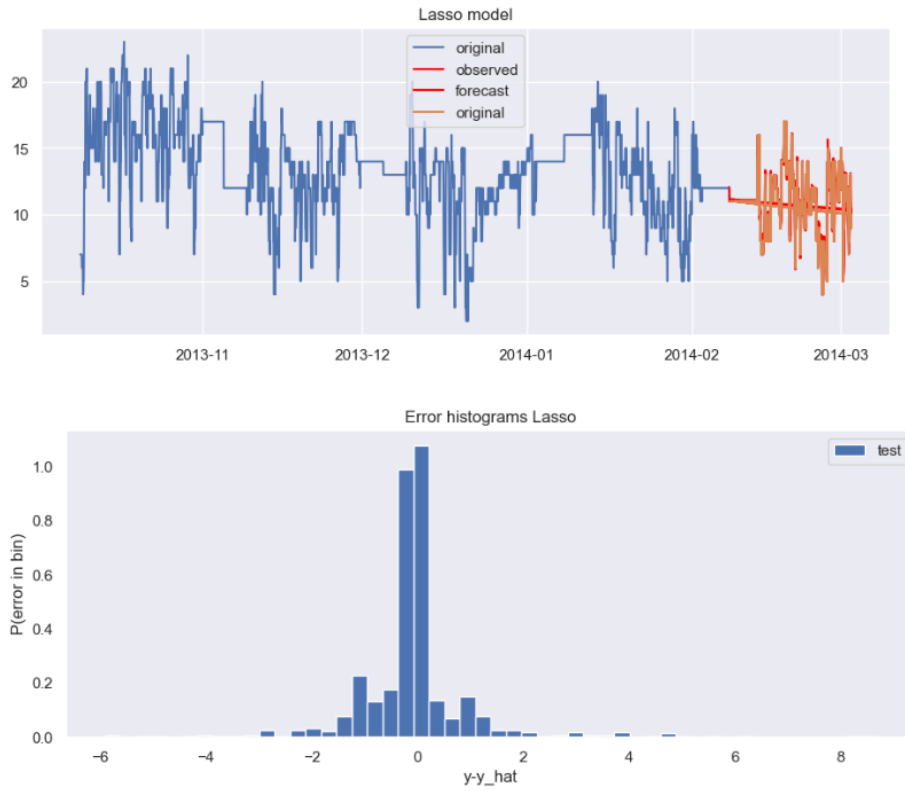
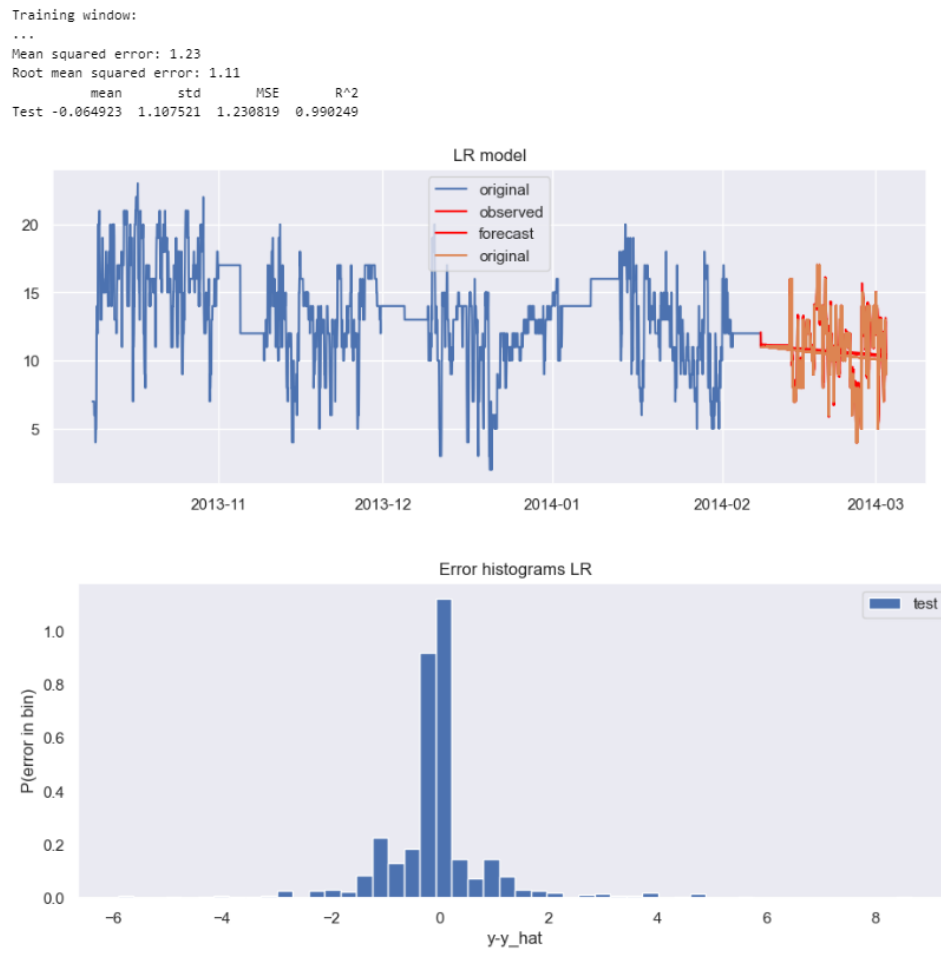


Figure 7.53: Lasso with horizon 48



**Figure 7.54:** Linear Regression with horizon 48

As can be seen, the Random Forest model continues to be the best performing.

## Chapter 8

# Conclusions and future works

The analysis of micromobility data in an urban setting using several machine learning models, including ARIMA, linear regression, Lasso, Ridge, Gradient Boosting, and Random Forest, was the focus of the thesis work described in this study. As can be seen, the methodology described was tested on two actual use cases from two distinct urban contexts. The conducted analyses and the results made it possible to write down some guidelines for using this framework, such as the best machine learning model to use, the range in which the value of the sliding window size or the value of the horizon must fall. In addition, this framework demonstrates that, depending on the situation, it may be necessary to enrich the data with new features, while in other situations it may not. Additionally, the results of this study may be helpful to companies that offer station-based bike sharing services in determining whether it will be necessary to move existing bikes to other stations or add new ones to the station under consideration. In the case of bike flow, it is important for businesses or municipalities to know whether there is sufficient demand to enable the development of useful services for the local community in that particular area.

Future research could build on this study by enriching the framework with more contextual metadata, such as the existence of points of interest nearby or the presence of specific events in the city context, all of which could improve the accuracy of the occupancy prediction. In order to profile users and tailor the service to their needs, user data from those who use it could also be integrated. Different mobility vectors might be explored, different cities and nations could use the same methodology, or the problem might be handled as a classification problem rather than a regression problem.

# Bibliography

- [1] *micromobilità in Vocabolario - Treccani* — *treccani.it*. [https://www.treccani.it/vocabolario/micromobilita\\_%28Neologismi%29/](https://www.treccani.it/vocabolario/micromobilita_%28Neologismi%29/). [Accessed 05-Apr-2023] (cit. on p. 4).
- [2] Horace Dediù. *The Micromobility Definition* — *micromobility.io*. <https://micromobility.io/news/the-micromobility-definition>. [Accessed 05-Apr-2023]. 2019 (cit. on p. 4).
- [3] Alexandre Santacreu. *Safe Micromobility*. Tech. rep. International Transport Forum, 2020 (cit. on pp. 4, 5, 7).
- [4] M. Ciuffini, S. Asperti, V. Gentili, R. Orsini, and L. Refrigeri. *6° rapporto nazionale sulla sharing mobility*. [Accessed 05-Apr-2023]. 2022. URL: <https://osservatoriosharingmobility.it/wp-content/uploads/2022/10/6-Rapporto-Nazionale-sharing-mobility.pdf> (cit. on p. 8).
- [5] Fabrizio Prati. *I sistemi di Bike sharing in Nord America*. [Accessed 05-Apr-2023]. 2020. URL: [https://www.lesscars.it/wp-content/uploads/2020/09/NACTO\\_bikesharing\\_prati\\_lesscars\\_low.pdf](https://www.lesscars.it/wp-content/uploads/2020/09/NACTO_bikesharing_prati_lesscars_low.pdf) (cit. on p. 8).
- [6] *Transport for London*. <https://cycling.data.tfl.gov.uk/>. [Online; accessed 19-January-2023] (cit. on p. 8).
- [7] *data.austintexas.gov*. <https://data.austintexas.gov/Transportation-and-Mobility/Shared-Micromobility-Vehicle-Trips/7d8e-dm7r/data>. [Online; accessed 19-January-2023] (cit. on p. 8).
- [8] *data.europa.eu*. <https://data.europa.eu/data/datasets/51661c47-30c4-4b4b-8c1d-f339c02b15fb?locale=it>. [Online; accessed 19-January-2023] (cit. on p. 8).
- [9] *Chicago data portal*. <https://data.cityofchicago.org/Transportation/E-Scooter-Trips-2020/3rse-fbp6/data>. [Online; accessed 19-January-2023] (cit. on p. 9).
- [10] *Norfolk Opendata*. <https://data.norfolk.gov/Government/Micromobility-Electric-Scooters-and-Bikes-/wqxq-hhe6>. [Online; accessed 19-January-2023] (cit. on p. 9).

- [11] *Fluctuo*. <https://dive.fluctuo.com/>. [Online; accessed 19-January-2023] (cit. on p. 9).
- [12] *Kaggle - SF Bay Area Bike Share*. <https://www.kaggle.com/datasets/benhamner/sf-bay-area-bike-share?resource=download&select=database.sqlite>. [Online; accessed 19-January-2023] (cit. on p. 10).
- [13] *Fremont Bridge Bicycle Counter | City of Seattle Open Data portal — data.seattle.gov*. <https://data.seattle.gov/Transportation/Fremont-Bridge-Bicycle-Counter/65db-xm6k>. [Accessed 19-January-2023] (cit. on p. 12).
- [14] *Did it rain in Seattle? (1948-2017)*. <https://www.kaggle.com/datasets/ratman/did-it-rain-in-seattle-19482017>. [Online; accessed 20-February-2023] (cit. on p. 12).
- [15] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. 1st ed. 2006. Corr. 2nd printing. Information science and statistics. Springer, 2006. ISBN: 9780387310732,0387310738 (cit. on pp. 17, 18).
- [16] Jamal Fattah, Latifa Ezzine, Zineb Aman, Haj El Moussami, and Abdeslam Lachhab. «Forecasting of demand using ARIMA model». In: *International Journal of Engineering Business Management* 10 (Jan. 2018), p. 184797901880867. DOI: 10.1177/1847979018808673. URL: <https://doi.org/10.1177/1847979018808673> (cit. on p. 17).
- [17] L.E. Melkumova and S.Ya. Shatskikh. «Comparing Ridge and LASSO estimators for data analysis». In: *Procedia Engineering* 201 (2017). 3rd International Conference “Information Technology and Nanotechnology”, ITNT-2017, 25-27 April 2017, Samara, Russia, pp. 746–755. ISSN: 1877-7058. DOI: <https://doi.org/10.1016/j.proeng.2017.09.615>. URL: <https://www.sciencedirect.com/science/article/pii/S1877705817341474> (cit. on pp. 18, 19).
- [18] Upma Singh, Mohammad Rizwan, Muhannad Alaraj, and Ibrahim Alsaidan. «A Machine Learning-Based Gradient Boosting Regression Approach for Wind Power Production Forecasting: A Step towards Smart Grid Environments». In: *Energies* 14.16 (2021). ISSN: 1996-1073. DOI: 10.3390/en14165196. URL: <https://www.mdpi.com/1996-1073/14/16/5196> (cit. on p. 19).
- [19] Nicholas E. Johnson, Olga Ianiuk, Daniel Cazap, Linglan Liu, Daniel Starobin, Gregory Dobler, and Masoud Ghandehari. «Patterns of waste generation: A gradient boosting model for short-term waste prediction in New York City». In: *Waste Management* 62 (Apr. 2017), pp. 3–11. DOI: 10.1016/j.wasman.2017.01.037. URL: <https://doi.org/10.1016/j.wasman.2017.01.037> (cit. on p. 20).

- [20] Alexei Botchkarev. «Evaluating Performance of Regression Machine Learning Models Using Multiple Error Metrics in Azure Machine Learning Studio». In: *SSRN Electronic Journal* (2018). DOI: 10.2139/ssrn.3177507. URL: <https://doi.org/10.2139/ssrn.3177507> (cit. on pp. 21, 22).
- [21] Alipujang Jierula, Shuhong Wang, Tae-Min OH, and Pengyu Wang. «Study on Accuracy Metrics for Evaluating the Predictions of Damage Locations in Deep Piles Using Artificial Neural Networks with Acoustic Emission Data». In: *Applied Sciences* 11.5 (2021). ISSN: 2076-3417. DOI: 10.3390/app11052314. URL: <https://www.mdpi.com/2076-3417/11/5/2314> (cit. on p. 23).
- [22] Jeremy Miles. *R Squared, Adjusted R Squared*. Sept. 2014. DOI: 10.1002/9781118445112.stat06627. URL: <https://doi.org/10.1002/9781118445112.stat06627> (cit. on p. 23).
- [23] Aryan Jadon, Avinash Patil, and Shruti Jadon. *A Comprehensive Survey of Regression Based Loss Functions for Time Series Forecasting*. 2022. arXiv: 2211.02989 [cs.LG] (cit. on p. 24).