Decoding Urban Mobility: Insights from bike rental data

Kaoyan, Chen   Joao Filipe, Rodrigues Dos Santos Natal Marques
Lizet Viviana, Silva Lizarro   Oussama, Touhami

2025-11-11

This project examines the factors that influence hourly bike-rental demand using a dataset that includes temporal, environmental, and contextual variables. Understanding these patterns is essential for designing smarter and greener urban mobility systems, as bike sharing helps reduce CO emissions, improve public health, and generate economic benefits. Through exploratory analysis, we identify the strongest predictors of bike usage and quantify their influence, providing insights that support efficient resource planning and enhance the sustainability of bike-sharing systems.

Introduction

Background and Motivation

We selected this topic aiming to explore the demand patterns of bike rental, because it directly contributes to smarter and greener city planning. By examining when and why people use shared bikes, the bike sharing company or the city can better allocate resources, reduce congestion, and promote eco-friendly mobility. From a data science perspective, the dataset presents diverse variable types and analytical challenges, making it a rich context for applying rigorous data cleaning and exploratory methods.

This scientific project is good for humanity because bike sharing delivers evidence-based societal benefits: it saves 46,000 tons of CO and 200 tons of air pollutants annually, helps prevent 1,000 chronic diseases, and leads to €40 million in healthcare savings each year. By shifting urban mobility away from personal cars, bike sharing eases congestion,saving 760,000 hours of productivity, which is valued at €30 million, and supports 6,000 full-time equivalent jobs all across Europe. Every euro invested in bike sharing offers at least 10% annual return in measurable positive externalities, making these systems highly efficient and sustainable. Learning about this topic is especially interesting because it combines environmental impact, public health improvement, and economic growth, all backed by recent large-scale data analyses.

Project Objectives

Our goals are:

1. Explore and visualize usage patterns on an hourly basis (e.g. by hour of day, day of week, season).
2. Build a regression model to estimate the number of bike rentals (cnt) given features such as weather, season, hour, and other contextual variables.
3. Interpret model results to understand which factors most strongly influence demand.
4. Provide insight and recommendations for operational decisions (for example, anticipating peak hours, adjusting supply).
5. More specifically, we aim to combine exploratory data analysis (EDA), explore the correlationship between dependent variables and independent variables, and create a model to predict the future trend, eventually to provide some managerial advice for the company.

Research Questions

1. What are the typical hourly demand patterns (e.g. morning peaks, evening peaks)?
2. How do weather conditions (temperature, humidity, wind, etc.) correlate with bike rentals?
3. Which features (hour, season, holiday, working day, weather) are the strongest predictors of demand?
4. How can insights from the model help in resource planning (e.g. pre-positioning bikes, increasing revenue, maintenance schedules)?

Data

Sources

Dataset: UCI Machine Learning Repository - Bike Sharing Dataset

Description

The dataset used in this project is the "Bike Sharing Dataset (hour.csv)", obtained from https://archive.ics.uci.edu/dataset/275/bike+sharing+dataset. hour.csv contains hourly records of bike rentals along with various temporal and environmental factors that influence demand. The data captures user activity across different weather conditions, seasons, and times of the day, providing a rich foundation for both exploratory analysis and predictive modeling.

Loading Data

```
Data loaded: (17379, 17)
```

	instant	dteday	season	yr	mnth	hr	holiday	weekday	workingday	weathersit	temp	ater
0	1	2011-01-01	1	0	1	0	0	6	0	1	0.24	0.28
1	2	2011-01-01	1	0	1	1	0	6	0	1	0.22	0.27
2	3	2011-01-01	1	0	1	2	0	6	0	1	0.22	0.27
3	4	2011-01-01	1	0	1	3	0	6	0	1	0.24	0.28
4	5	2011-01-01	1	0	1	4	0	6	0	1	0.24	0.28

Wrangling

General Transformations

Document the data preprocessing steps taken, including cleaning, transformation, and any merging of datasets.

```
Confirm that the type is datetime64[ns] and that years 2011-2012 appear as expected:
Converted data type: datetime64[ns]
Unique years in dataset: [2011 2012]
```

	dteday	hr	hour_group	weekday_name	season_name	is_weekend	weather_name
0	2011-01-01	0	Night	Saturday	Winter	1	Clear
1	2011-01-01	1	Night	Saturday	Winter	1	Clear
2	2011-01-01	2	Night	Saturday	Winter	1	Clear
3	2011-01-01	3	Night	Saturday	Winter	1	Clear
4	2011-01-01	4	Night	Saturday	Winter	1	Clear
5	2011-01-01	5	Night	Saturday	Winter	1	Mist

	dteday	hr	hour_group	weekday_name	season_name	is_weekend	weather_name
6	2011-01-01	6	Morning	Saturday	Winter	1	Clear
7	2011-01-01	7	Morning	Saturday	Winter	1	Clear
8	2011-01-01	8	Morning	Saturday	Winter	1	Clear
9	2011-01-01	9	Morning	Saturday	Winter	1	Clear
10	2011-01-01	10	Morning	Saturday	Winter	1	Clear
11	2011-01-01	11	Morning	Saturday	Winter	1	Clear
12	2011-01-01	12	Afternoon	Saturday	Winter	1	Clear
13	2011-01-01	13	Afternoon	Saturday	Winter	1	Mist
14	2011-01-01	14	Afternoon	Saturday	Winter	1	Mist
15	2011-01-01	15	Afternoon	Saturday	Winter	1	Mist
16	2011-01-01	16	Afternoon	Saturday	Winter	1	Mist
17	2011-01-01	17	Afternoon	Saturday	Winter	1	Mist
18	2011-01-01	18	Evening	Saturday	Winter	1	Light Snow/Rain
19	2011-01-01	19	Evening	Saturday	Winter	1	Light Snow/Rain
20	2011-01-01	20	Evening	Saturday	Winter	1	Mist
21	2011-01-01	21	Evening	Saturday	Winter	1	Mist
22	2011-01-01	22	Evening	Saturday	Winter	1	Mist
23	2011-01-01	23	Evening	Saturday	Winter	1	Mist

```
Encoding complete. New shape: (17379, 32)
```

	instant	dteday	season	yr	mnth	hr	holiday	weekday	weathersit	temp	...	hour_group_
0	1	2011-01-01	1	0	1	0	0	6	1	0.24	...	False
1	2	2011-01-01	1	0	1	1	0	6	1	0.22	...	False
2	3	2011-01-01	1	0	1	2	0	6	1	0.22	...	False
3	4	2011-01-01	1	0	1	3	0	6	1	0.24	...	False
4	5	2011-01-01	1	0	1	4	0	6	1	0.24	...	False

	count	mean	std	min	25%	50%	75%	max
temp	17379.0	0.496987	0.192556	0.02	0.3400	0.5000	0.6600	1.0000
atemp	17379.0	0.475775	0.171850	0.00	0.3333	0.4848	0.6212	1.0000
hum	17379.0	0.627229	0.192930	0.00	0.4800	0.6300	0.7800	1.0000
windspeed	17379.0	0.190098	0.122340	0.00	0.1045	0.1940	0.2537	0.8507

Summary of Transformations

Step	Transformation	Justification	Validation
1	Load dataset	Import raw data into DataFrame	Confirmed shape (17k × 16)
2	Convert `dteday` to datetime	Enable time-based analysis	Checked data type & years
3	Add derived features (`season_name`, `is_weekend`, `hour_group`, `wheather_name`, ‘weekday_name’)	Improve interpretability & grouping	Reviewed sample outputs
4	Encode categorical variables	Prepare data for ML	Verified shape and columns
5	Check feature ranges	Validate normalized columns	Confirmed 0–1 range consistency

---

All transformations are **justified**, **documented**, **reproducible**, and **validated**, ensuring a transparent preprocessing workflow.

```
(17379, 21)
```

	instant	dteday	season	yr	mnth	hr	holiday	weekday	weathersit	temp	...	hum	windsp
0	1	2011-01-01	1	0	1	0	0	6	1	0.24	...	0.81	0.0
1	2	2011-01-01	1	0	1	1	0	6	1	0.22	...	0.80	0.0
2	3	2011-01-01	1	0	1	2	0	6	1	0.22	...	0.80	0.0
3	4	2011-01-01	1	0	1	3	0	6	1	0.24	...	0.75	0.0
4	5	2011-01-01	1	0	1	4	0	6	1	0.24	...	0.75	0.0

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17379 entries, 0 to 17378
Data columns (total 21 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   instant       17379 non-null  int64
 1   dteday        17379 non-null  datetime64[ns]
 2   season        17379 non-null  int64
 3   yr            17379 non-null  int64
 4   mnth          17379 non-null  int64
```

```
 5   hr             17379 non-null  int64
 6   holiday        17379 non-null  int64
 7   weekday        17379 non-null  int64
 8   weathersit     17379 non-null  int64
 9   temp           17379 non-null  float64
 10  atemp          17379 non-null  float64
 11  hum            17379 non-null  float64
 12  windspeed      17379 non-null  float64
 13  casual         17379 non-null  int64
 14  registered     17379 non-null  int64
 15  cnt            17379 non-null  int64
 16  season_name    17379 non-null  category
 17  weekday_name   17379 non-null  category
 18  is_weekend     17379 non-null  object
 19  hour_group     17379 non-null  category
 20  weather_name   17379 non-null  category
dtypes: category(4), datetime64[ns](1), float64(4), int64(11), object(1)
memory usage: 2.3+ MB
```

	instant	dteday	season	yr	mnth	hr
count	17379.0000	17379	17379.000000	17379.000000	17379.000000	17379.000
mean	8690.0000	2012-01-02 04:08:34.552045568	2.501640	0.502561	6.537775	11.546752
min	1.0000	2011-01-01 00:00:00	1.000000	0.000000	1.000000	0.000000
25%	4345.5000	2011-07-04 00:00:00	2.000000	0.000000	4.000000	6.000000
50%	8690.0000	2012-01-02 00:00:00	3.000000	1.000000	7.000000	12.000000
75%	13034.5000	2012-07-02 00:00:00	3.000000	1.000000	10.000000	18.000000
max	17379.0000	2012-12-31 00:00:00	4.000000	1.000000	12.000000	23.000000
std	5017.0295	NaN	1.106918	0.500008	3.438776	6.914405

Spotting Mistakes and Missing Data

```
Missing values per column:
 instant         0
dteday          0
season          0
yr              0
mnth            0
hr              0
holiday         0
weekday         0
```

```
weathersit      0
temp            0
atemp           0
hum             0
windspeed       0
casual          0
registered      0
cnt             0
season_name     0
weekday_name    0
is_weekend      0
hour_group      0
weather_name    0
dtype: int64
Remaining missing values: 0

=== BASIC DOMAIN CHECKS ===
hr in [0,23]: True
mnth in [1,12]: True
weekday in [0..6] (0=Sunday..6=Saturday): True
yr in {0,1}: True

=== CONSISTENCY WITH dteday ===
mnth matches dteday.month: True
weekday(Sun0) matches dteday: True
yr (0->2011, 1->2012) matches dteday.year: True

=== HOURLY COVERAGE PER DAY (expect 0..23 each day) ===
Days with missing hours: 76
dteday
2011-01-02       [5]
2011-01-03    [2, 3]
2011-01-04       [3]
2011-01-05       [3]
2011-01-06       [3]
Name: hr, dtype: object

Days with extra/out-of-range hours: 0

 No mismatches found between encoded variables and dteday (for checked columns).

hr
```

```
24    655
23     62
22      6
18      2
16      1
12      1
8       1
17      1
1       1
11      1
Name: count, dtype: int64
------------------------------------------
2011-01-02 00:00:00: missing hours [5]
2011-01-03 00:00:00: missing hours [2, 3]
2011-01-04 00:00:00: missing hours [3]
2011-01-05 00:00:00: missing hours [3]
2011-01-06 00:00:00: missing hours [3]
2011-01-07 00:00:00: missing hours [3]
2011-01-11 00:00:00: missing hours [3, 4]
2011-01-12 00:00:00: missing hours [3, 4]
2011-01-14 00:00:00: missing hours [4]
2011-01-18 00:00:00: missing hours [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
2011-01-19 00:00:00: missing hours [3]
2011-01-22 00:00:00: missing hours [5]
2011-01-23 00:00:00: missing hours [4]
2011-01-24 00:00:00: missing hours [2]
2011-01-25 00:00:00: missing hours [3]
2011-01-26 00:00:00: missing hours [3, 4, 18, 19, 20, 21, 22, 23]
2011-01-27 00:00:00: missing hours [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
2011-01-28 00:00:00: missing hours [4]
2011-01-29 00:00:00: missing hours [5]
2011-01-30 00:00:00: missing hours [6]
2011-02-01 00:00:00: missing hours [4]
2011-02-03 00:00:00: missing hours [4]
2011-02-04 00:00:00: missing hours [4]
2011-02-09 00:00:00: missing hours [4]
2011-02-10 00:00:00: missing hours [3]
2011-02-11 00:00:00: missing hours [3, 4]
2011-02-13 00:00:00: missing hours [5]
2011-02-15 00:00:00: missing hours [3]
2011-02-16 00:00:00: missing hours [2]
2011-02-20 00:00:00: missing hours [5]
2011-02-22 00:00:00: missing hours [0, 1, 2, 3, 4, 5]
```

```
2011-02-23 00:00:00: missing hours [4]
2011-02-24 00:00:00: missing hours [4]
2011-02-25 00:00:00: missing hours [4]
2011-02-27 00:00:00: missing hours [5]
2011-02-28 00:00:00: missing hours [2, 4]
2011-03-06 00:00:00: missing hours [5]
2011-03-07 00:00:00: missing hours [2]
2011-03-10 00:00:00: missing hours [3, 4]
2011-03-11 00:00:00: missing hours [4]
2011-03-13 00:00:00: missing hours [2]
2011-03-14 00:00:00: missing hours [4]
2011-03-15 00:00:00: missing hours [3]
2011-03-16 00:00:00: missing hours [2]
2011-03-18 00:00:00: missing hours [4]
2011-03-21 00:00:00: missing hours [4]
2011-03-23 00:00:00: missing hours [4]
2011-03-27 00:00:00: missing hours [5]
2011-03-28 00:00:00: missing hours [4]
2011-04-11 00:00:00: missing hours [3]
2011-08-27 00:00:00: missing hours [18, 19, 20, 21, 22, 23]
2011-08-28 00:00:00: missing hours [0, 1, 2, 3, 4, 5, 6]
2011-09-06 00:00:00: missing hours [1]
2011-09-08 00:00:00: missing hours [2]
2011-09-12 00:00:00: missing hours [3]
2011-10-19 00:00:00: missing hours [3]
2011-11-28 00:00:00: missing hours [2]
2011-12-25 00:00:00: missing hours [4]
2011-12-26 00:00:00: missing hours [3]
2011-12-28 00:00:00: missing hours [4]
2012-01-02 00:00:00: missing hours [3]
2012-01-10 00:00:00: missing hours [3]
2012-01-17 00:00:00: missing hours [3]
2012-02-06 00:00:00: missing hours [3]
2012-02-20 00:00:00: missing hours [4]
2012-02-21 00:00:00: missing hours [3]
2012-02-29 00:00:00: missing hours [4]
2012-03-11 00:00:00: missing hours [2]
2012-04-02 00:00:00: missing hours [3]
2012-04-11 00:00:00: missing hours [4]
2012-10-29 00:00:00: missing hours [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 1
2012-10-30 00:00:00: missing hours [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
2012-11-08 00:00:00: missing hours [3]
2012-11-29 00:00:00: missing hours [3]
```

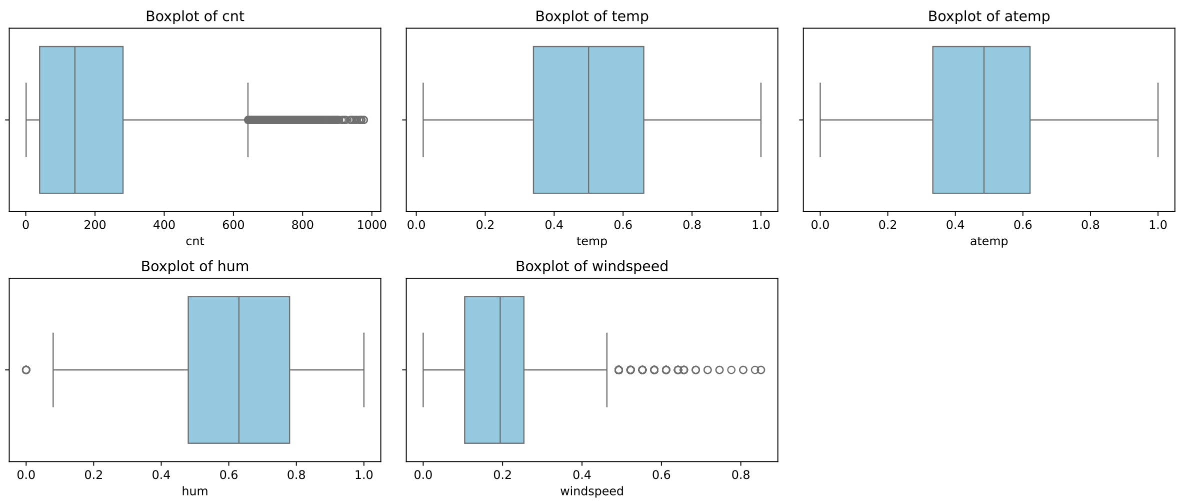
```
2012-12-24 00:00:00: missing hours [4]
2012-12-25 00:00:00: missing hours [3]
```

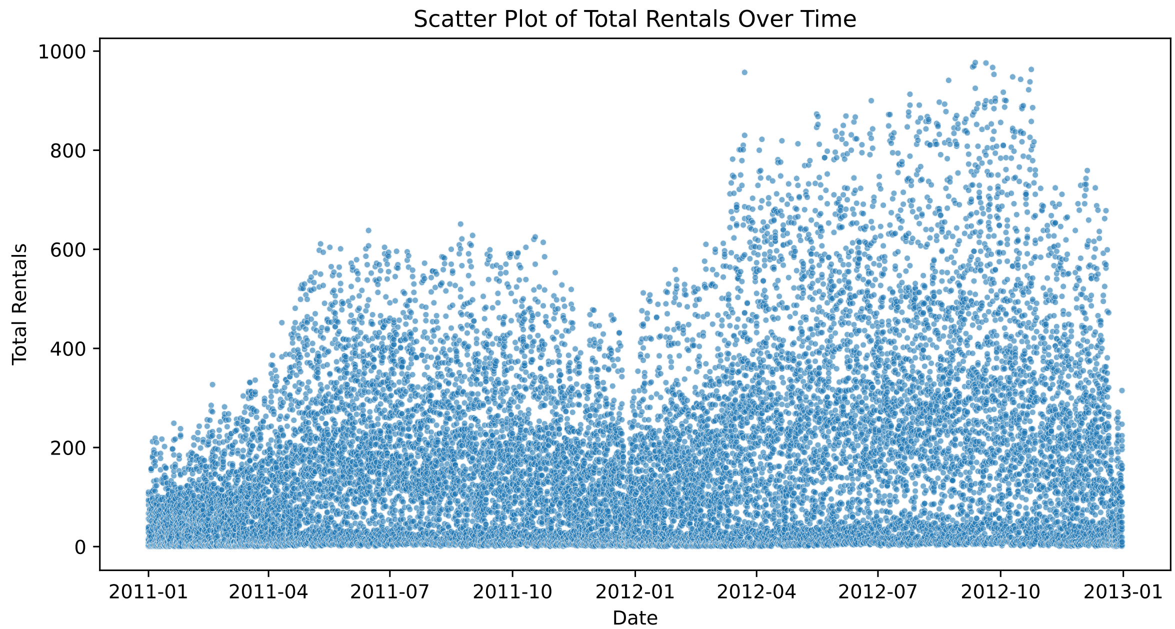
Listing Anomalies and Outliers

This section identifies and analyzes anomalies or outliers in the **Bike Sharing Dataset**. Outlier detection is important for understanding data quality, variability, and rare events.

Outliers are evaluated using: - **Visual inspection:** box plots and scatter plots
- **Statistical methods:** Interquartile Range (IQR) and Z-scores
- **Domain knowledge:** contextual judgment (e.g., rental counts, weather extremes)

Not all outliers are removed — some represent real and meaningful phenomena (e.g., holidays, weather shocks).





Interpretation of Boxplots

The boxplots above visualize the spread and potential outliers among the main numerical variables (`cnt`, `temp`, `atemp`, `hum`, and `windspeed`).

- **`cnt` (total rentals):** Shows several high-end points beyond the upper whisker, representing **peak demand hours** (e.g., weekday commutes or summer weekends). These are not data errors but **valid extreme values**.

- **`temp` and `atemp`:** Both are normalized between 0 and 1, showing smooth distributions without extreme deviations, suggesting well-scaled temperature measures.

- **`hum` (humidity):** Displays a slightly skewed distribution with upper-end values near 1.0, likely corresponding to humid or rainy days — realistic rather than anomalous.

- **`windspeed`:** Contains a few high-end outliers, possibly due to **sensor measurement spikes** or rare weather conditions. These are few and not impactful on model training.

Conclusion:
Most outliers represent **real phenomena** rather than data-entry errors.
Thus, they will be **retained** for further exploratory analysis and modeling to preserve the natural variability of bike rental behavior.

##Data Cleaning Checklist Before moving to analysis, verify:

All column names are clean and consistent Data types are appropriate for each variable Missing values are identified and handled Outliers are investigated and documented Categorical variables are properly encoded Duplicate rows are checked and removed if needed Date/time variables are in proper format Cleaned data is saved for reproducibility

```
Column names:
['instant', 'dteday', 'season', 'yr', 'mnth', 'hr', 'holiday', 'weekday', 'weathersit', 'temp

Data types:
instant                  int64
dteday          datetime64[ns]
season                   int64
yr                       int64
mnth                     int64
hr                       int64
holiday                  int64
weekday                  int64
weathersit               int64
temp                   float64
atemp                  float64
hum                    float64
windspeed              float64
casual                   int64
registered               int64
cnt                      int64
season_name           category
weekday_name          category
is_weekend              object
hour_group            category
weather_name          category
dtype: object

Missing values per column:
instant         0
dteday          0
season          0
yr              0
mnth            0
hr              0
holiday         0
weekday         0
weathersit      0
```

```
temp            0
atemp           0
hum             0
windspeed       0
casual          0
registered      0
cnt             0
season_name     0
weekday_name    0
is_weekend      0
hour_group      0
weather_name    0
dtype: int64

Outlier summary:
Extreme rental values already checked - valid high peaks retained.

Categorical and datetime validation:
Categorical columns: ['is_weekend']
Datetime column: datetime64[ns]

Duplicate rows count: 0
Date range: 2011-01-01 00:00:00 → 2012-12-31 00:00:00

 Cleaned dataset saved successfully as 'bike_cleaned.csv'
```

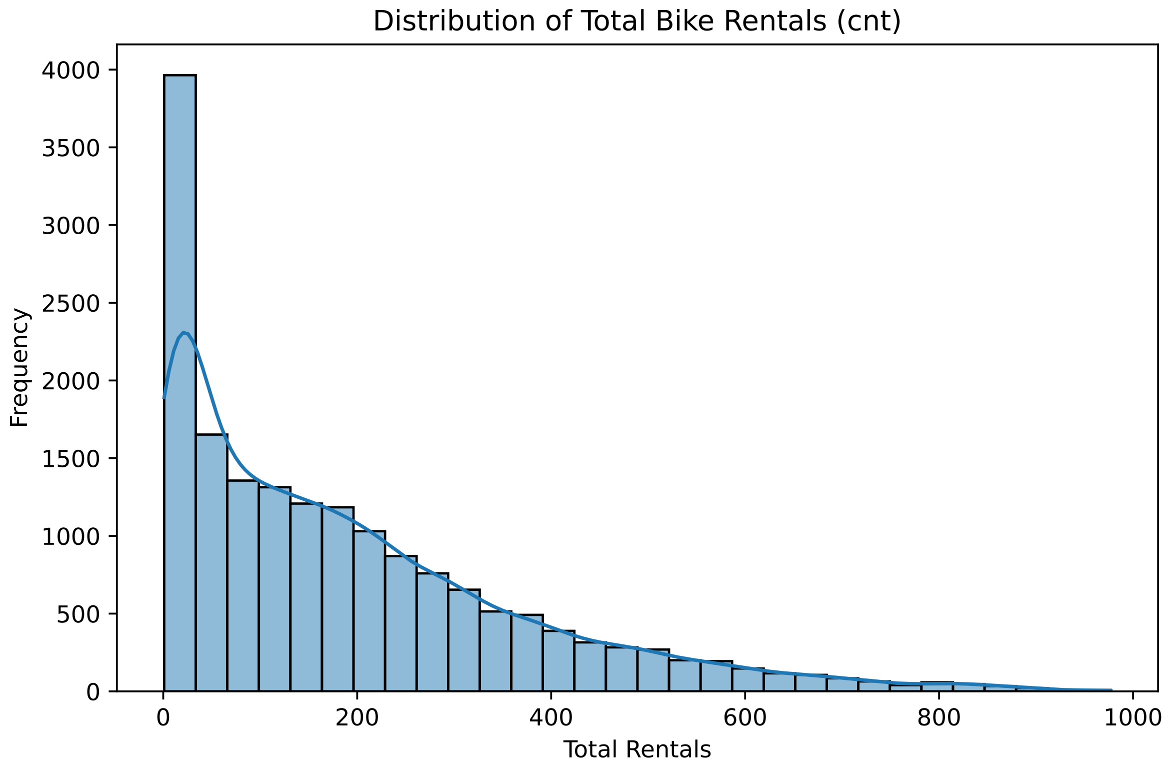
EDA – Exploring Variable Distributions

We start by examining how individual variables behave (their variation). Following Module 3, we visualize both **numeric** and **categorical** variables to understand their shape, spread, and possible outliers.

Hourly Demand Patterns Analysis

Understanding the behavior of the bike-sharing system requires an analytical approach that considers multiple temporal scales. The analysis begins with the Distribution of Bike Rentals to provide an overview of overall usage levels, and then moves to the examination of hourly, daily, and monthly patterns, which reveal variations across the day, week, and year. A subsequent analysis of seasonal trends helps clarify how climatic conditions shape demand throughout the year. Finally, contrasting hourly patterns with seasonal effects allows us to understand how short-term and long-term temporal dynamics interact to define overall system usage.

Distribution of Bike Rentals

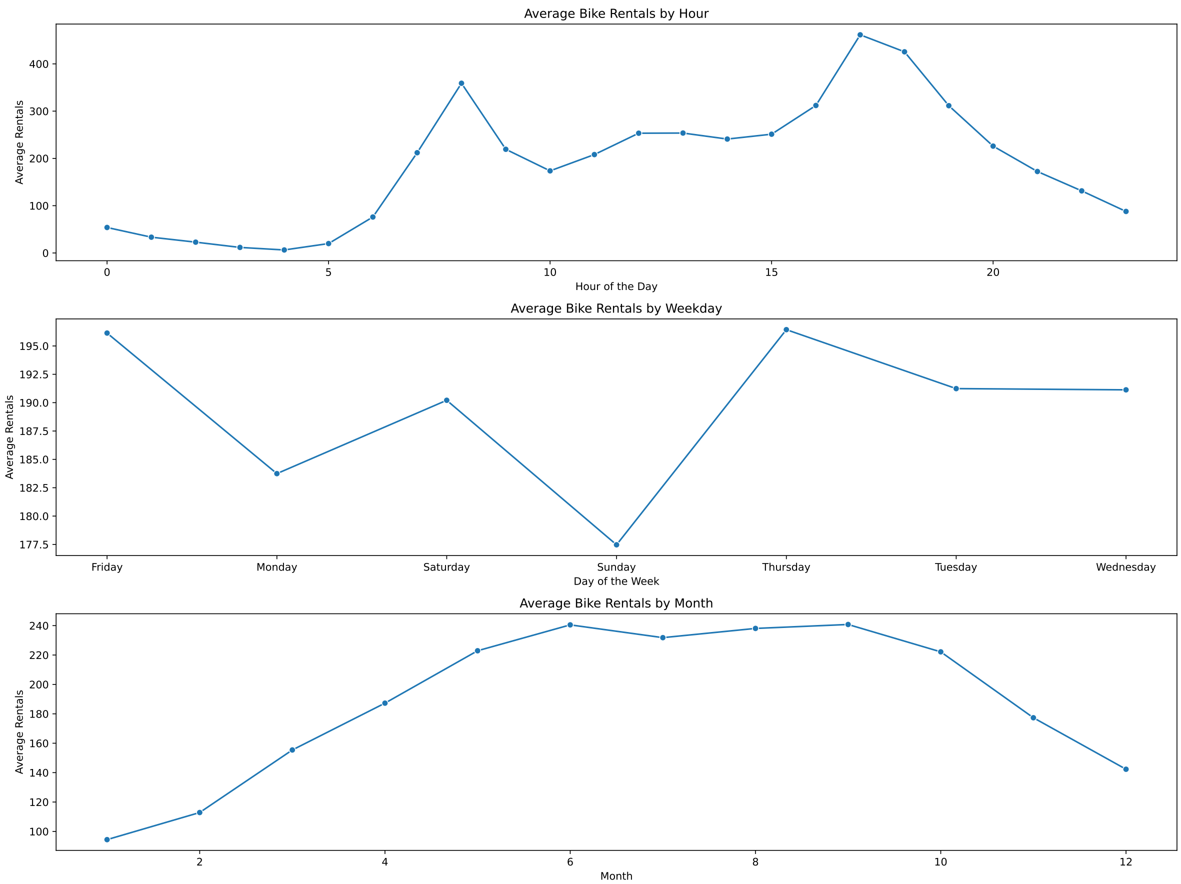


The distribution of total bike rentals (cnt) is strongly right-skewed, indicating that most hours record relatively low to moderate rental activity, while a smaller number of hours reach very high demand peaks.

This pattern suggests that: - Bike usage is not constant throughout the day, there are specific times (likely rush hours or weekends) with much higher demand. - The skewness reflects contextual influences such as weather, working days, and time of day. - The majority of hours have fewer than 200 rentals, showing that high-demand situations (over 600 rentals) are exceptional events rather than the norm.

Overall, the shape highlights high variability and strong temporal effects in bike-sharing demand, justifying deeper exploration by time and external conditions.

Variations across the day, week, and year



Average Bike Rentals by Hour

The hourly pattern shows two clear peaks one around 8 AM and another near 5–6 PM, corresponding to morning and evening commuting hours. Early morning periods (before 6 AM) and late night hours (after 9 PM) show minimal activity. This behavior indicates that bike sharing is primarily used for work- or school-related travel, reflecting strong time-of-day dependencies and typical urban mobility rhythms.

Average Bike Rentals by Weekday

Bike rentals tend to be higher on working days, particularly Thursday and Friday, and lower on weekends, especially Sunday. This pattern reinforces the idea that the system is mainly used for functional weekday transportation rather than leisure. The moderate usage observed on Saturdays suggests some recreational use but at a lower intensity compared to weekdays.

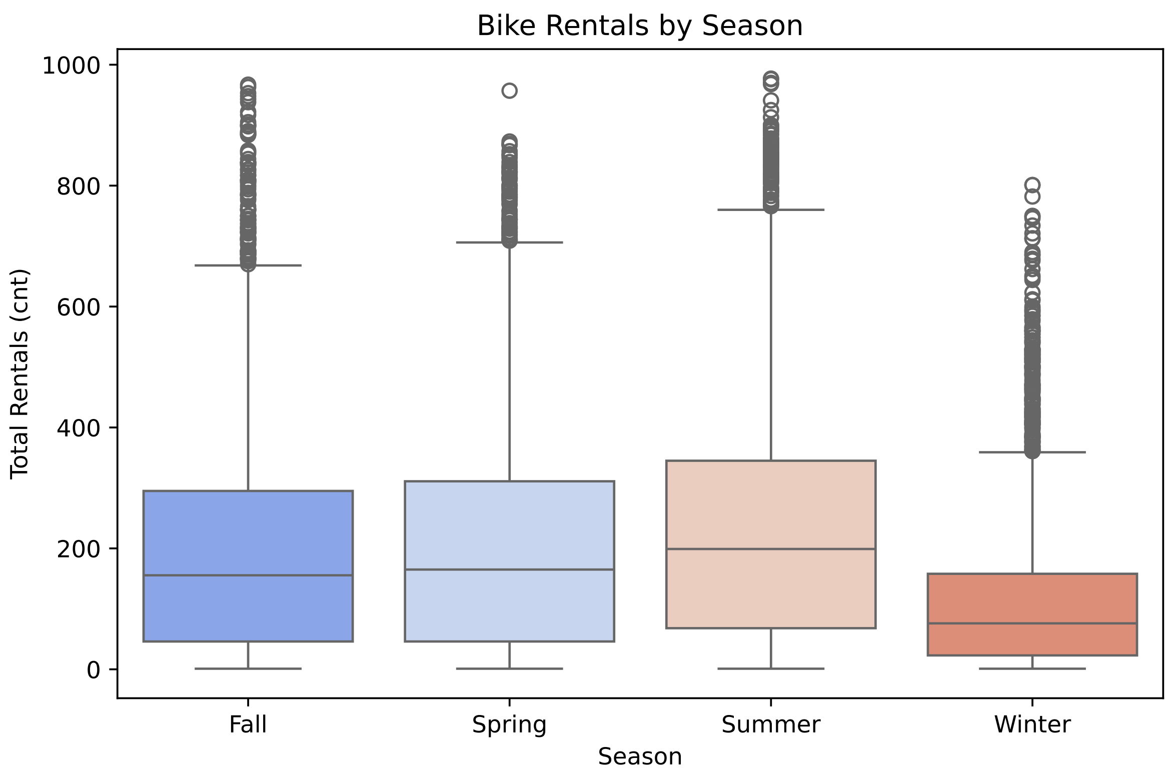
Average Bike Rentals by Month

The monthly pattern reveals a strong seasonal trend: rentals increase steadily from February to June, remain high throughout summer (June–September), and decline sharply from October onward. This reflects the influence of weather and temperature, as warmer conditions generally promote cycling, while colder months reduce usage due to less favorable conditions.

Taken together, these visualizations show that bike rental demand follows clear temporal cycles: daily (commuting peaks), weekly (higher weekday usage), and seasonal (weather-driven fluctuations). Understanding these patterns is essential for resource allocation, system planning, and operational decision-making.

Demand by Season

Boxplot of total rentals by season



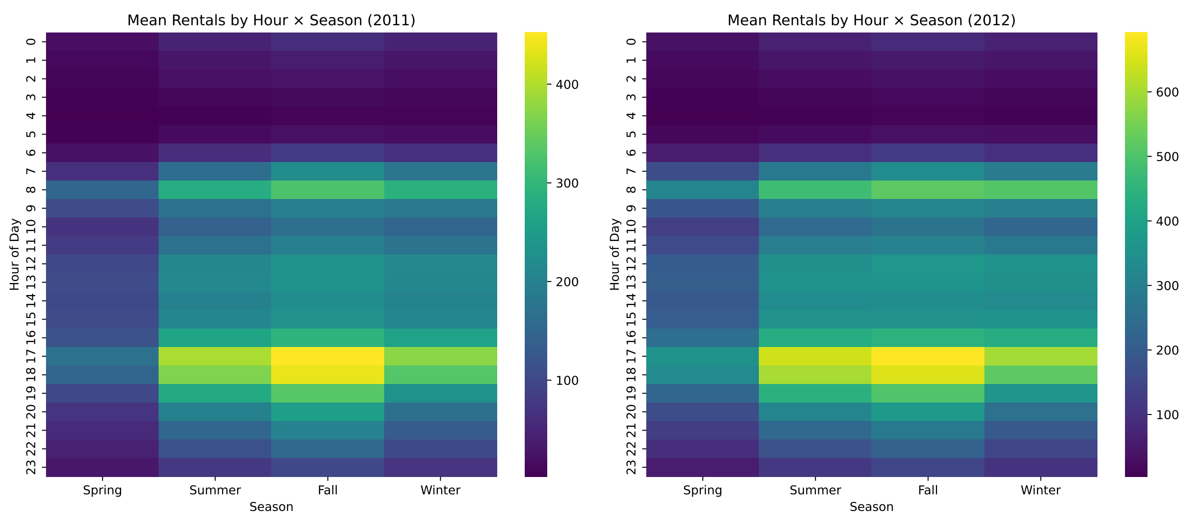
The boxplot shows clear seasonal differences in total bike rentals: - Summer and Fall display the highest median rental counts and the widest interquartile ranges (IQR), indicating both higher and more variable demand. - Spring presents moderate rental levels, reflecting the gradual return of favorable weather. - Winter has by far the lowest median and spread, showing that cold and harsh conditions drastically reduce bike usage.

The presence of outliers in all seasons — especially in Summer and Fall — suggests that certain peak days or hours experience unusually high demand, possibly due to special events or ideal weather.

Bike rental activity is strongly seasonal, peaking in warm months and dropping sharply in winter, confirming that weather and temperature are major drivers of usage in bike-sharing systems.

Interaction Between Hourly Patterns and Seasonal Variations

Building on these findings, it becomes essential to examine not only how demand varies across seasons but also how seasonal conditions influence the hourly distribution of rentals. By comparing hourly patterns within each season, we can determine whether the characteristic morning and evening peaks observed throughout the year intensify, weaken, or shift depending on weather conditions. This combined perspective allows for a deeper understanding of how short-term (hourly) and long-term (seasonal) temporal dynamics interact, and it sets the stage for the comparative analysis that follows.

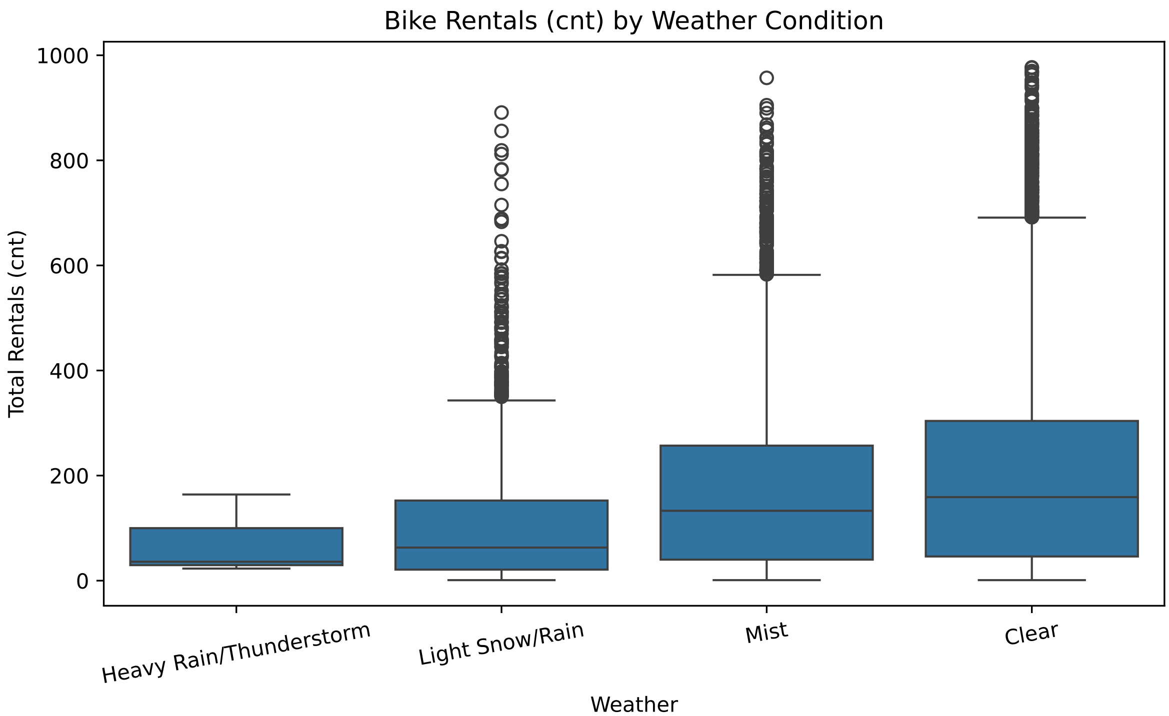


Insight	2011	2012	Interpretation
Peak hours	8 AM, 5–6 PM	Same	Commuting-driven demand
Highest seasons	Summer & Fall	Same but stronger	Weather is the dominant factor
Overall demand	Moderate	Much higher	System growth/adoption
Winter usage	Low	Low	Weather constraints persist

The heatmaps show that bike-sharing demand follows clear and stable daily cycles, with strong peaks during morning and evening commuting hours. Summer and fall consistently display the highest usage, while winter remains the lowest. Demand is notably higher in 2012, indicating system growth and wider user adoption. These patterns highlight the importance of planning fleet allocation and operations around predictable peak periods.

How Weather Conditions Affect Bicycle Rental Demand

Impact of Weather Conditions

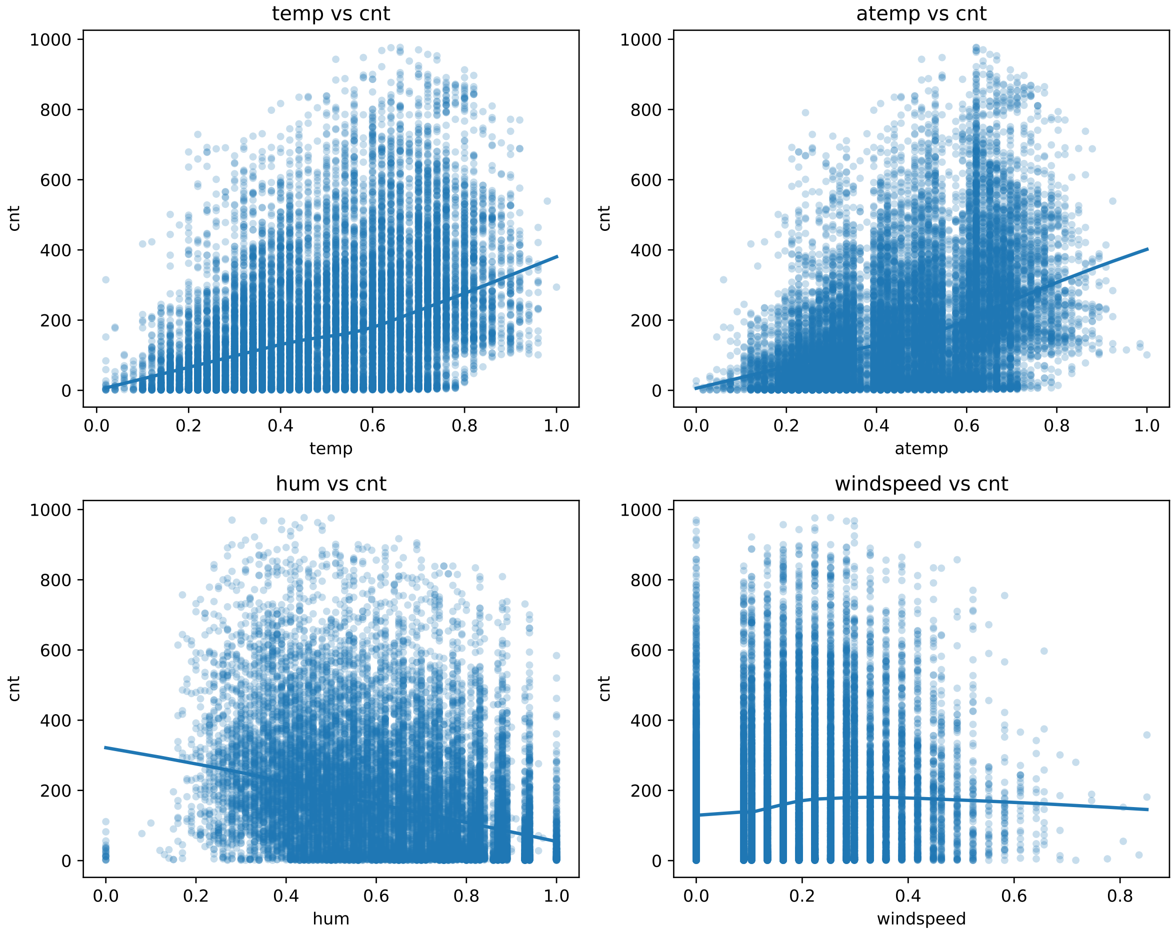


The boxplot clearly shows that bike rentals decrease as weather worsens:

Under clear weather, rentals are the highest and most variable. Mist and light snow/rain conditions show moderate but visibly lower usage. During heavy rain or thunderstorms, bike usage drops sharply, with very low median and narrow spread.

This confirms that adverse weather strongly discourages cycling, while clear conditions maximize ridership.

Effect of Environmental Factors on Bike Rental Demand

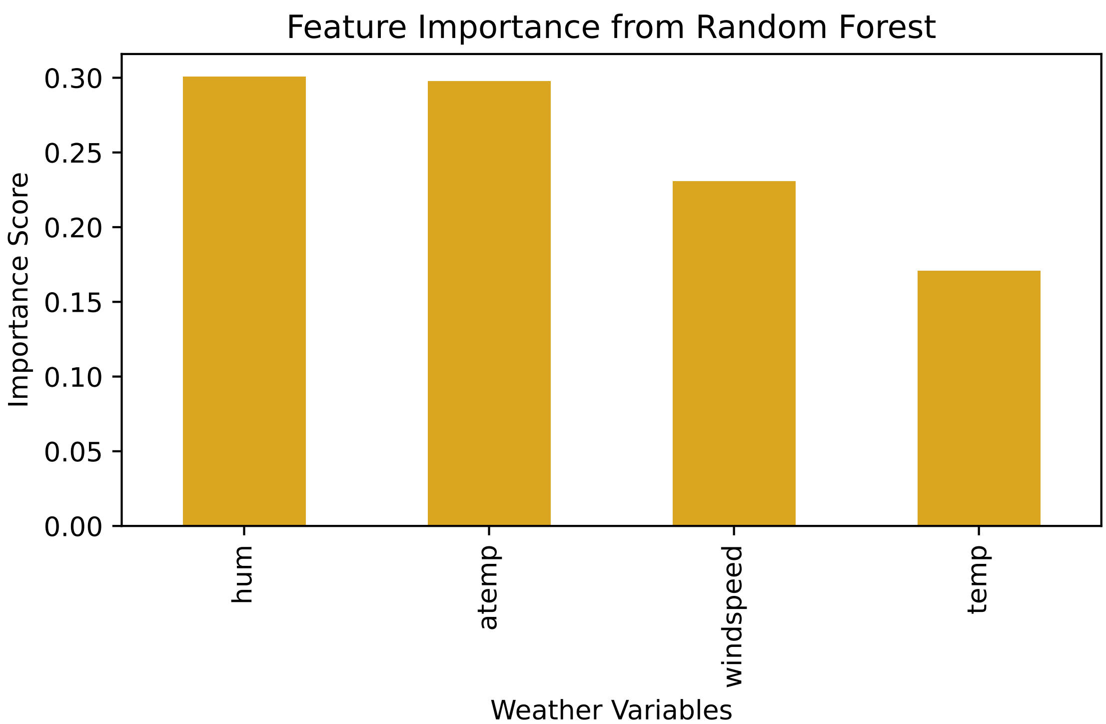


Variable	Relationship	Interpretation
Temperature (temp)	++ Strong Positive	Higher temperatures increase bike usage, especially in comfortable ranges.
Feels-like temp (atemp)	+ Positive	Pleasant perceived temperature encourages cycling.
Humidity (hum)	− Slight Negative	High humidity reduces usage due to discomfort or rain-related conditions.
Windspeed (windspeed)	− Weak Negative	Stronger winds make biking less appealing, though the effect is small.

Effect of Environmental Factors on Bike Rental Demand

To strengthen the understanding of how weather conditions shape bike rental behavior, it is helpful to use a modeling approach that can reveal the relative contribution of each climatic variable. The Random Forest Regressor offers this advantage by evaluating which features most effectively improve prediction accuracy across numerous decision trees. The resulting importance scores provide actionable insights, highlighting which environmental factors such as humidity, perceived temperature, windspeed, or actual temperature

```
hum          0.300848
atemp        0.297703
windspeed    0.230750
temp         0.170698
dtype: float64
```



Preliminary Analysis

Before choosing a final prediction model, we first carried out an exploratory analysis to better understand how the variables behave and how they relate to bike rental demand. At this stage, the goal is not to build the strongest predictive model yet, but to identify patterns that will guide our modelling decisions.

Methods Used So Far

1. Exploratory Data Analysis (EDA)

We mainly used visual tools such as:

- **Heatmaps** to explore how demand changes across hours and seasons.
- **Boxplots** to compare distributions of rentals across weather conditions and seasons.
- **Scatterplots and trend lines** to observe relationships between environmental variables (temperature, humidity, windspeed) and demand.
- **Distribution plots** to study the shape and skewness of rental counts.

EDA helps us understand the structure of the data, detect patterns, confirm assumptions, and identify which variables are likely to be important later.
For example, strong hourly peaks and seasonal differences suggest temporal variables will play a key role.

Next Steps Toward Modelling

Although we explored the data visually, we have not yet selected the final model. However, based on our findings, we will consider:

Potential models to test:

- **Multiple Linear Regression** (simple, interpretable baseline)
- **Random Forest Regression** (can capture patterns not seen in EDA)

These models are not final choices yet; they are candidates informed by what we learned through EDA.

So far, our analysis has focused on: - Understanding temporal and weather-related patterns - Identifying variables that seem important based on visual exploration - Preparing the data for modelling (cleaning, encoding, feature creation)

In the next stage, we will test and compare predictive models using the insights gained from EDA.

Appendix A: Description of Variables in hour.csv

A.1 Temporal Variables

Variable	Description
dteday	Date
mnth	Month (1–12)
hr	Hour of the day (0–23)
weekday Day of the week	(0 = Sunday … 6 = Saturday)
workingday	1 = Working day, 0 = Weekend or holiday
holiday	Indicates if the day is a holiday

A.2 Seasonal and Weather Category Variables

Variable	Description
season	Season (1 = Spring, 2 = Summer, 3 = Fall, 4 = Winter)
weathersit	Categorical weather situation

Code	Weather Description
1	Clear, few clouds, partly cloudy
2	Mist + cloudy / mist + broken clouds / mist + few clouds
3	Light snow; light rain + thunderstorm + scattered clouds; light rain + scattered clouds
4	Severe weather: heavy rain + ice pellets + thunderstorm + mist; snow + fog

A.3 Continuous Weather Variables

Variable	Description
temp Normalized temperature	(actual temp / 41°C)
atemp Normalized "feels-like" temperature	(feels-like temp / 50°C)
temp hum Normalized humidity	(humidity / 100)
windspeed Normalized wind speed	(windspeed / 67)

A.4 Demand Variables

Variable	Description
casual	Rentals by non-registered users
registered	Rentals by registered users
cnt Total	number of bike rentals (casual + registered)