POLISH-JAPANESE ACADEMY
OF INFORMATION TECHNOLOGY

**Computer Science**

**Informatyka**

Data Science

**Szymon Wujec**
S20431

# “Data analysis in terms of resource usage based on “Bike Sharing” dataset”

Master Thesis

Thesis supervisor
Dr. habil. Grzegorz Marcin Wójcik

Warsaw, June, 2024

## Table of contents

# 1 Introduction

## 1.1 Thesis structure description

The main goal of this thesis is to analyze a prepared set of data using various techniques or available tools in order to better understand what the target customers of urban bicycle rental companies are driven by or what conditions are conducive to better business development in this service industry. I was prompted to write such a paper by my own experience, I myself very often use such a means of transport as a bicycle in my free time. Whenever I find myself in front of a city bike sharing station and intend to rent a bike to get from point "A" to point "B", I am curious to know how a company providing bikes for public use plans when and where it should provide an increased number of these vehicles, and where it can afford to have fewer. For such a process and business to function properly and to be efficient, an in-depth analysis of usage data is required, which I intend to do in this thesis.

## 1.2 Goals of the thesis

The main purpose of analyzing the "Bike sharing" dataset on urban bicycle rentals is to understand rental patterns, determine the factors affecting the amount of urban bicycles rented by users. This makes it possible to forecast demand, segment users, assess performance and the impact of external factors, and most importantly, the profitability of the business.

## 1.3 Main thesis issues

### 1.3.1 Bike - Sharing systems

There are many programs commonly referred to as "urban bicycles" around the world. This colloquial name has a lot to do with the real state of affairs because a significant portion of the organizations providing services in the bicycle rental industry are actually city boards paid for with taxpayer money. Of course, over the years as this market significantly began to grow, through increased traffic, air pollution and even the magnification of the problem of global warming, private companies began to appear on the market offering such services. The location of bicycles for rent, as well as the occurrence and possibility of bike-sharing programs are usually used in large urban areas. At first they occurred in the close center of such cities, while currently we can notice them in locations where public traffic is much more intense. Such locations include, among others, subway stations, since the subway is undoubtedly the urban means of transport that has the largest number of transported passengers, while other popular locations are facilities where access by car is much more difficult. Let's imagine that in a certain city there is currently a football match taking place, the teams having a huge number of fans. It is obvious that the stadium would sell out all available tickets long before the start of the match, by the increased interest in the event. The largest stadium in the world is able to accommodate 114,000 people in the audience, if everyone wanted to come in their own car, the space allocated for parking would have to occupy 137 hectares, almost seven times the size of the stadium itself, assuming that the average area needed to park a passenger car is 12 square meters. In such situations and places, "city bike" stations are great, generating much less space, not to mention pollution.

**Picture 1 Bike-sharing station**

Bike - Sharing systems can be divided into five generations of these programs.

### 1.3.1.1 Generation zero

Stations that have a bike rental service belong to the zero generation of Bike - Sharing systems. They are characterized by the fact that such locations or stations are not automated, but run by employees or volunteers. This is undoubtedly the oldest rental model available in the world, currently the most recognized rental companies using this model are the network of sports equipment stores "Decathlon".

Picture 2 Bike-Sharing generation 0

### 1.3.1.2 First generation

The first instance of an unmanned rental service took place in Amsterdam in 1965. Dutch industrial designer Luud Schimmelpennink, decided with his friends to collect 50 bicycles, repaint them white, and then set them up around the city for free public use. This program was dubbed the "White Bicycle Plan." Unfortunately, most of these bicycles were stolen.

Picture 3 Left side, Luud Schimmelpennink Picture 4 Right side, "White Bicycle Plan"

### 1.3.1.3 Second generation

The authors of this model are Morten Sadolina and Ole Wessung. They developed a bicycle rental model that involved free rental of a bicycle vehicle in exchange for a deposit. This deposit was in the form of coins, which then unlocked access to the bikes, a similar system to the one we know from hypermarkets and their shopping carts. The first pool of accessible bicycles was located in Denmark and consisted of 26 transport units along with 4 stations between 1991 and 1993. The next big step in 1995 was the introduction of 800 transport units in Copenhagen, and this system was called "Bycyklen."

Picture 5 Bike-Sharing second generation

#### 1.3.1.4 Third generation

Generation three consists of docking stations where you can rent a bicycle vehicle and then drop it off at any station belonging to the same Bike-Sharing system. The stations are equipped with stands, which have mechanisms for releasing and locking bicycles left there only with a computer system. Bicycle rental is based on the identification of the registered person with a membership program card. Such a rental model was developed by Hellmut Slachta and Paul Brandstätter between 1990 and 1992 as "Public Velo," while it was first implemented under the name "Bikeabout" in 1996 by the University of Portsmouth and Portsmouth City Council in England.

Picture 6 Bike-Sharing Third generation

#### 1.3.1.5 Fourth generation

The generation in question has its two variants. The first is in fact a rental-ready bicycle of the fourth generation, which is one where the bicycles have their own docking stations as in the second or third generation, while the bicycle vehicles are equipped with locks that allow the user to decide independently whether to leave the bicycle at a dedicated docking station or park it anywhere by using the equipped lock. On the other hand, the aforementioned second variant of generation four are bicycles covered by bike-sharing programs called generation five. The difference is that the variant called generation five does not have dedicated docking stations, but has only the locks already known from the first variant of generation four. The entire fourth generation was developed by the German railroad and logistics company, "Deutsche Bahn" in 1998 to use automatically generated digital authentication codes for automated locking and unlocking of bicycles. Their proprietary system in 2000 was called "Call a Bike," which relied on unlocking a bicycle using an SMS message or phone call, to later restructure and launch a fully operational application to be installed on a phone. In Poland, the first programs with fourth-generation bikes in their fleet appeared in 2015 in Krakow under the name "Wavelo" and in 2017 in Warsaw with the name "Acro-bike"

**Picture 7 Bike-Sharing Fourth generation**

(Bicycle-sharing system, 2024)

### 1.3.2 Analysis

Analysis can be called the process when we break a complex subject into its first factors or smaller parts in order to ultimately draw specific conclusions and gain a better understanding of the issue at hand. Analysis is a word dating from around 1500 to 300 years before the birth of Christ from Ancient Greek, which was "analusis" and meant "breaking up" or "an untying." Through this range on the timeline we could see a division into four periods; the "Mycenaean" Greek which lasted between 1500-1200 years before Christ, the "Dark Ages" whose time was between 1200-800 before Christ, the "Archaic or Epic period" in effect between 800-500 before Christ, to finally end with the Classical period which lasted the shortest, beginning between 500 and 300 years before the birth of Christ. This technique officially and under its name was used for learning mathematics, solving logical problems, while this process was used much earlier but under an unspecified form by any thinking beings living in the universe. An example is hunting or defending from being hunted, the process of choosing the right place where an individual should be in order to hunt prey, or choosing a safe and secluded place is also a process of strategic analysis. On the other hand, the term "deconstructive analysis" or "critical analysis" is very common in literature and art. These statements involve searching for meanings ultimately hidden by authors or artists. This type of analysis scans a work of art in order to understand the deeper meaning or significance hidden under the veil of transference, censorship, or many other similar ways of masking the deeper meaning of the artist's work. There are many types or varieties of analysis, it all depends on what we subject to the analysis process in question.

(Analysis, 2024)

### 1.3.2.1 Data Analysis

The term "Data Analysis," as the name implies, refers to the processing of data in order to better understand it and, based on it, draw accurate conclusions. Elements used in this process can include inspection, transformation, modeling and even data cleaning. Carrying out this type of analysis helps in decision-making, for which it is necessary to take into account a number of criteria. To perform such analysis it is necessary to follow several steps, which are strictly enforced, for example, by companies providing analytical services. The first step, which has an impact on the final result, which is the result of the analysis performed, is to determine the requirements needed to complete the required range of data that will later form a set of data. For example, if the end user, i.e. the person commissioning the analysis, expects specific conclusions on the subject of the bicycle industry, then in the dataset we can expect to see data such as models of bicycles, their weights, sizes and many others. On the other hand, we should not expect the appearance of information not relevant to the commissioned analysis, in this case it could be, for example, medical data, which have little to do with the subject of bicycles. After determining the requirements for what information the dataset should contain, data collection follows based on the requirements established in the previous step. The data should be collected from as many sources as possible, in order to avoid biasing the result of the data analysis. The data can take various forms, from textual, going through numerical data, and ending with graphical data. The next step is the stage called "Data processing". Data processing consists in arranging the collected data resulting from the previous stages, in such a way as to enable analysis. For example, if the analysis concerns numerical data, it is necessary to place the data in rows and columns in a table, then we talk about structured data. Data structuring usually occurs in data storage programs created for analytical or statistical purposes, among others, such as a program of the "Microsoft" form of the "Office" package called "Excel". Data cleaning is another very important point in preparing data for analysis, as the collected data may contain numerous duplicates and even errors, which at a later stage may have a huge impact on the outcome of the analysis, making it impossible to reach the right conclusions of the analysis performed. For example, at the time of compiling the financial data needed to analyze the portfolio of the client ordering the analysis, there may be duplicates of goods held, while in reality they exist in a single quantity. Such a mistake can result in falsely inflated financial information about the assets held, and then lead to legal problems, such as those related to tax issues. The final stage of analysis can be done by means of modeling and algorithms, which are usually mathematical equations that help to better understand the issue and the thesis of the collected data. On the other hand, with current technology, machine learning programs are most often used to better screen the collected data set and then provide accurate conclusions based on the analysis.

(Data analysis, 2024)

### 1.3.3 Machine learning

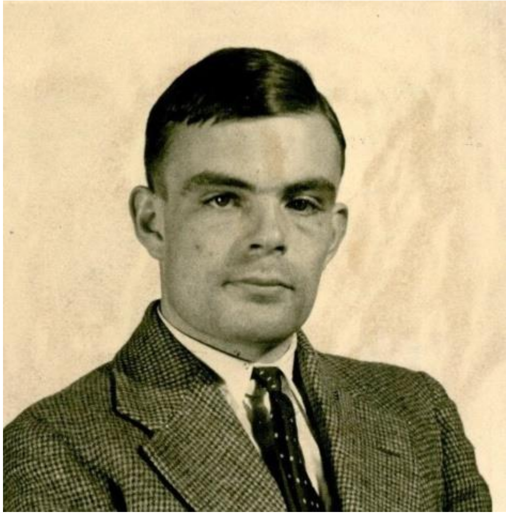
Machine learning, often called using the shortcut "ML," is the branch of specialization that is artificial intelligence, which is also very often called using the shortcut "AI." ML has the task of performing a "thought" process with the help of various algorithms and AI, which is trained on the basis of a provided data set in textual, numerical or graphical form. In traditional programming, a human writes all the rules for the computer to follow. In machine learning, the computer creates these rules itself by analyzing examples. To begin with, in order to train a model, a dataset containing data related to the problem we want to solve will be required. An example might be a model with which we want to recognize cats, in which case we should prepare a compilation of a large number of photos of cats and other animals. The next step is to train the model, which learns from the provided data. During training, the model analyzes the data and tries to find patterns that help it recognize what is depicted in the photos. After the training process, you can proceed to testing. Testing is done on new data that the model has not seen before, to see how well it finds itself in recognizing the assumed objects using the patterns it finds. The teaching process can be likened to teaching a child by showing him objects such as a banana, apple, orange, emphasizing the name of the depicted thing. The child should find specific patterns, such as that bananas are usually yellow and oblong or apples are red and round. The problem may arise if we keep presenting a red apple, and at the time of the test we show an apple with a green color. Therefore, in the process of completing the dataset, it is very important that the data be different, if we present pictures of cats, there must not be only the same cat. Once the model is trained and tested, and the results are satisfactory, the model is ready to be used in practice.

(Oxford, 2019)

#### 1.3.3.1 History of Machine Learning

The person we can call the originator of artificial intelligence is Alan Turing. In 1950, Turing proposed the Turing test as a way to test a machine to see if it could think like a human. The test involves a kind of game called the "imitation game." The experiment involves 3 characters, a human, a machine and a judge. The judge is the person who initiates and conducts the test by asking questions via an interface, which can be, for example, a text chat. The interface is essential in this experiment because the judge does not know which character he is talking to is a human and which is a machine. After asking a question and receiving an answer, the judge analyzes the answers, and after the series of questions is done, the result of the Turing test occurs, that is, the moment when the judge indicates unambiguously which character is a human and which is a machine. On the other hand, if the judge is unable to unambiguously indicate, the machine passes the Turing test. The purpose of such a test was to assess the ability of machines to demonstrate intelligence. Turing believed that if a machine can carry on a conversation in a manner indistinguishable from that of a human, then the machine can be considered intelligent.

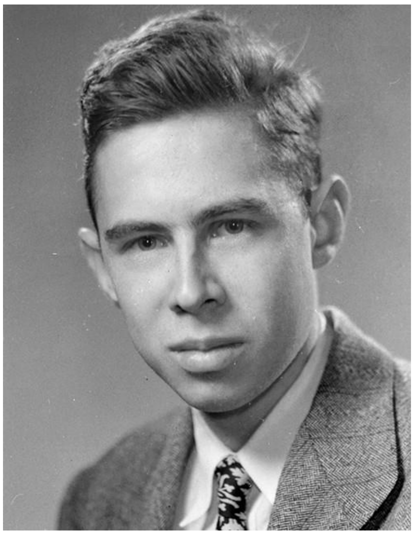
(Alan Turing, 2024)



**Picture 8 Left side, Alan Truing Picture 9 Right side, Working Alan Turing**

In 1957, Frank Rosenblatt created the perceptron, the first machine learning algorithm inspired by neurons in the human brain. It was an early model of a neural network. Rosenblatt's work on the perceptron was groundbreaking because it showed for the first time that machines could learn and adapt based on experience. Although the classical perceptron had its limitations, such as its inability to solve non-linear problems, an example is the XOR problem, which is a logical operation that returns a true result only when one of the two inputs is true, unfortunately Rosenblatt's Perceptron, a simple neural network model, only works well for problems that can be solved with one simple line. Since the XOR problem is not linearly separable, the perceptron cannot solve it correctly. Nevertheless, the classical perceptron became the foundation for the development of more advanced neural network architectures. Frank Rosenblatt's contribution to the development of artificial intelligence is crucial, and his research on the perceptron has formed the foundation for the development of modern machine learning methods that are revolutionizing our approach to solving complex problems in various fields of science and technology.

(Frank Rosenblatt, 2024)



**Picture 10 Frank Rosenblatt**

One of the first recorded computer programs that can learn from its experiences is a checkers program by Arthur Samuel. Arthur is an American engineering pioneer in the field of computer technology and the use of AI. The aforementioned program of his, he developed in 1959. The program used a technique called "self-play," playing thousands of games against itself. This allowed him to analyze different strategies and learn which moves were best in different situations. Samuel introduced algorithms and evaluation functions that allowed the program to judge how good a position was on the board. The program used these ratings to make better decisions during the game. His ideas about self-learning computers were revolutionary and inspired many future researchers. Arthur Samuel is often called one of the fathers of machine learning. His innovative approach to self-learning computers and the game of checkers had a huge impact on the development of AI. Today, his works are still cited and studied as fundamental texts in the field of artificial intelligence and machine learning.

(Arthur Samuel, 2024)

> *“As a result of these experiments one can say with some certainty that it is now possible to devise learning schemes which will greatly outperform an average person and that such learning schemes may eventually be economically feasible as applied to real-life problems.”*
>
> (Samuel, 1959)

**Picture 11 Arthur Samuel**

All of the above-mentioned personalities and their achievements or discoveries lead us to the times in which we are currently living. On a daily basis, machine learning is widely used in various fields, from medicine to marketing. Every now and then news circulates about the release for public use of a new model of the widely known "Chat GPT" by the company "OpenAI". Each of the developments brings humanity closer to advanced artificial intelligence.

The history of machine learning is a story of gradual development, algorithms and, consequently, technology. Over the years, data collected, can be analyzed automatically and lead to new conclusions and scientific discoveries. From the early work of Alan Turing and Frank Rosenblatt, to the development of neural networks, to modern deep learning techniques, machine learning is constantly evolving and gaining importance in our daily lives.

(Foote, 2021)

### 1.3.4 **Refactoring**

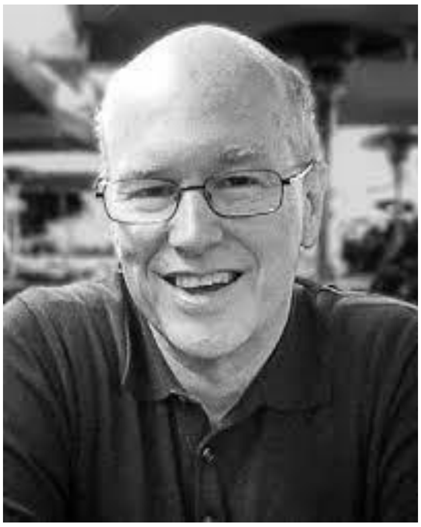
Refactoring is a technique used to reorganize existing code without changing its functionality. Programmers often use this method to avoid repetitive code by creating universal methods with variables. This makes the code clearer, more readable and more compact, making it easier to further develop and maintain. Refactoring also helps find and eliminate bugs, leading to more stable and efficient software. The term "refactoring" was first used in 1990 by William Opdyke and Ralph Johnson in their published work. Since then, the technique has gained immense popularity and has become a standard tool in the arsenal of every experienced programmer. Martin Fowler, a British software developer, has shared his extensive industry knowledge through various publications and books. His works focus on best practices in programming and software design. One notable book is "Refactoring: Improving the Design of Existing Code," published in 2002 in collaboration with William Opdyke. The book has become a key guide for programmers around the world, offering practical tips and techniques for refactoring. In his publications, Flower often emphasizes the importance of refactoring in the context of agile methodologies such as Agile, which promote iterative software development. Refactoring is integral to these methodologies, as it enables regular improvements in code quality without having to stop work on new functionality.

(Code refactoring, 2024)

> *“Help in understanding the code also helps me spot bugs. I admit I'm not terribly good at finding bugs. Some people can read a lump of code and see bugs, I cannot. However, I find that if I refactor code, I work deeply on understanding what the code does, and I put that new understanding right back into the code. By clarifying the structure of the program, I clarify certain assumptions I've made, to the point at which even I can't avoid spotting the bugs. 49 It reminds me of a statement Kent Beck often makes about himself, "I'm not a*

*great programmer; I'm just a good programmer with great habits." Refactoring helps me be much more effective at writing robust code.”*

(Flower, Beck, Opdyke, & Brant, 2002)



**Picture 12 Left side, Martin Flower Picture 13 Right side, William Opdyke**

# 2 Description of analysis

## 2.1 Description introduction

The analysis of the "Bike sharing" dataset was carried out in two approaches. The first approach was to screen the "Bike sharing" dataset manually, using tools such as the "Excel" program from the "Office" package by "Microsoft". Then, using the analytical formulas available in this program that allow transformations, formatting, modeling of data, leading to interesting results, the final conclusions were drawn. On the other hand, the second approach is to leave the analysis of the data contained in the "Bike sharing" dataset to a program from the field of Machine Learning, which generates graphs and extracts all the interesting resultant data in a way that is understandable to the user. These analyses provide a better understanding of when and why bicycles are rented, which can be useful for bike rental managers to better plan and manage their resources.

## 2.2 Manual analysis description

The analysis of the data from the "day.csv" file will be divided into several steps, with a brief description of each analysis, the method of visualization and potential conclusions that can be drawn.

### 2.2.1 Seasonal analysis

To begin with, the number of bicycle rentals in different seasons will be compared. The bar chart prepared will show how many times bicycles were rented in spring, summer, autumn and winter. It will then be possible to see in which months rentals are most frequent, which may be due to better weather at certain times of the year.

### 2.2.2 Annual analysis

Comparisons will be made between rentals in two different years: 2011 and 2012. A bar chart will show the total number of rentals for each year. It will be possible to see whether the popularity of bike rentals is increasing or decreasing. If the number of rentals in 2012 is higher, it can be assumed that the service is gaining in popularity.

### 2.2.3 Impact of working days and holidays

It will examine how the number of rentals differs between weekdays and weekends and holidays. A bar chart will show the differences. It can be expected that on weekdays there are fewer rentals because people are at work, while on weekends and holidays there are probably more rentals because of more free time.

### 2.2.4 Weather influence

The number of bicycle rentals under different weather conditions will be compared. A bar chart will show how different types of weather affect rentals. It will be possible to see if there are more or fewer rentals on sunny and cloudless days than on rainy days, which will help understand the impact of weather on user behavior.

### 2.2.5 Monthly analysis

It will examine how the number of rentals changes by month of the year. A line graph will show this change. It may be that there are more rentals in summer and spring than in winter, which may be related to better weather and higher temperatures.

### 2.2.6 Analysis by days of the week

It will analyze how the number of rentals varies by day of the week. A bar chart will show how many bikes were rented on each day of the week over two years. It will be possible to see if the number of rentals could be influenced, for example, by the fact that users commute to work if there were more rentals during the working week than weekend days.

### 2.2.7 Comparison of registered and casual users

The number of rentals by registered and occasional users will be compared. A bar chart will show the differences between these groups. It will be possible to see if registered users rent bicycles more often, which could be due to lower costs per rental or other benefits for such users.

### 2.2.8 The impact of holidays on rentals

A thesis will be posed for analysis as to whether the number of rentals differs on holidays. A bar chart will show the differences. It could be that there are fewer rentals on holidays because people are spending time with family, or those rentals will be more because they have more free time for recreation.

## 2.3 Automated analysis description

Data analysis using Machine Learning is the harnessing of currently available technology to perform complex analysis in a relatively short period of time, comparing to analysis performed manually by a human. Of course, one has to keep in mind the level of difficulty and intricacy of the analysis being performed, as there are situations where the human dominates the machine and technology. However, when analyzing a "Bike sharing" dataset that contains information stretched over two years, technology makes the analysis colorful and sometimes even deeper. The process of such automated analysis foresees several key steps, and each step is indispensable to reach the final step of drawing conclusions of the analysis.

### 2.3.1 Data upload

The data is loaded from a file with the extension "CSV" into a program written in Python. The data relates to bicycle rentals and includes information on various characteristics, such as temperature, humidity, day of the week and the like.

### 2.3.2 Data preparation

The data is prepared for analysis. Column values, such as temperature, are scaled to be in the correct units. Then columns that are not needed for analysis, such as the date column, are removed.

### 2.3.3 Modeling

In this step, linear regression is used. It is used to determine how different characteristics, such as temperature, humidity, day of the week and the like, affect the number of bikes rented. The model is trained on the entire data set, which allows a more accurate estimation of the effect of individual characteristics on the number of rented bicycles. What's more, this approach, by training the model on the entire dataset, enables a better understanding of all the relationships between features.

### 2.3.4 Model results

After training the model, regression coefficients are analyzed. These values indicate how strongly each characteristic affects the number of bicycles rented. For example, a high coefficient for temperature means that temperature has a strong influence on bicycle rentals. In addition, the results are presented in the form of positive and negative numbers, which makes it possible to determine the relationship of the change in the values of the characteristics with their progressive or regressive influence on the number of rented bicycles. For example, an increase in the coefficient saying that the weather conditions have worsened will probably get a negative result of the regression coefficient. One should treat this information in such a way that a decrease in weather conditions reduces the number of rented bicycles, and the larger the absolute value of such a regression coefficient, the characteristic assigned to this coefficient, much more influences, even dictates the number of rented bicycles.

### 2.3.5 Visualization of results

Graphs are created to help interpret the results. A correlation matrix shows how different traits are related to each other. "Heatmap" visualizes these correlations, allowing you to see which traits are strongly related to each other and which are less so. Additional graphs such as bar or dot plots are most closely used to visualize the correlation of one of the traits with the interpreted trait, which is the number of rented bicycles.

### 2.3.6 Chart analysis

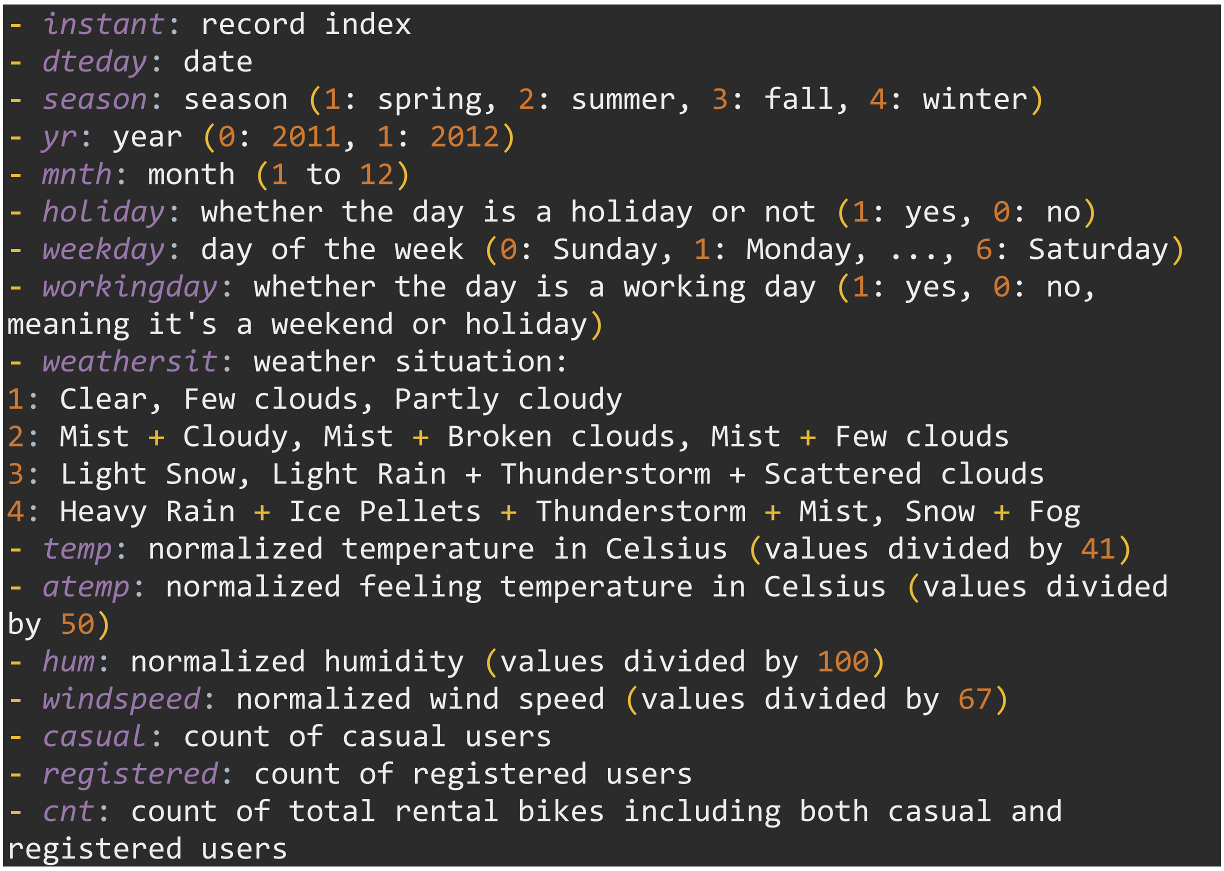
The last stage, which would not have been possible to reach without performing the previous ones, is the analysis of the generated graphs. The conclusions of such an analysis are an important moment of any analysis, because a wrong understanding of the resultant values, or making mistakes at the stages of data preparation or modeling, can carry serious consequences. For example, if the result of such an analysis were to influence a restructuring of strategy or even a re-branding of the company commissioning the analysis, this would carry enormous, unnecessarily generated costs. In order to compile results showing a broader view of the relationship between the characteristics in the dataset, a correlation matrix works well. It allows you to see how different factors affect each other. For example, you can simultaneously see that the number of rented bicycles is strongly correlated with temperature, and temperature is strongly correlated with season. Point charts, on the other hand, show the relationship between a particular characteristic and the number of rented bicycles. Using them, it is possible to show whether bikes are rented more often on warmer days.

## 2.4 Dataset

The dataset relates to the "Capital Bikeshare" bicycle rental system in Washington D.C., U.S.A., and includes historical data from 2011-2012. Bicycle rental systems provide automatic bicycle rental and return, and data from these systems can be used to analyze mobility in the city. In addition, the data can help in studies on the impact of various environmental factors on bicycle rentals.

Picture 14 Capital Bikeshare rental bicycle

Data on rented bicycles of the company "Capital Bikeshare" from 2011-2012 can be found in a file using the extension "csv" under the name "day.csv". Each entry in this data set represents another day in the two-year range. In effect, we are using a dataset that has 731 entries, and each entry has 16 values. These values are named in the first line.



```
- instant: record index
- dteday: date
- season: season (1: spring, 2: summer, 3: fall, 4: winter)
- yr: year (0: 2011, 1: 2012)
- mnth: month (1 to 12)
- holiday: whether the day is a holiday or not (1: yes, 0: no)
- weekday: day of the week (0: Sunday, 1: Monday, ..., 6: Saturday)
- workingday: whether the day is a working day (1: yes, 0: no,
meaning it's a weekend or holiday)
- weathersit: weather situation:
1: Clear, Few clouds, Partly cloudy
2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds
3: Light Snow, Light Rain + Thunderstorm + Scattered clouds
4: Heavy Rain + Ice Pellets + Thunderstorm + Mist, Snow + Fog
- temp: normalized temperature in Celsius (values divided by 41)
- atemp: normalized feeling temperature in Celsius (values divided
by 50)
- hum: normalized humidity (values divided by 100)
- windspeed: normalized wind speed (values divided by 67)
- casual: count of casual users
- registered: count of registered users
- cnt: count of total rental bikes including both casual and
registered users
```

Figure 1 List of attributes in dataset “Bike sharing”

The author of the dataset is Hadi Fanaee-T, while the owner is the "Laboratory of Artificial Intelligence and Decision Support (LIAAD), University of Porto, INESC Porto." It is a research unit affiliated with the University of Porto, one of the largest and most prestigious universities in Portugal. "LIAAD is known for its research in artificial intelligence and decision support systems. Decision support systems include decision analysis and risk management, business intelligence, optimization techniques, and simulation and modeling, among others. The goal of "LIAAD" is to advance social knowledge and technology in the fields of artificial intelligence and decision support systems.

(LIAAD, 2024)

Picture 15 Hadi Fanaee-T

The dataset in question has guidelines so that if used in publications it must be cited according to the publication presented below.

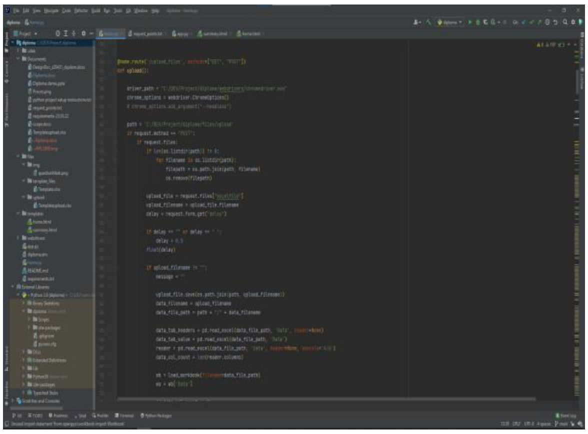
"Fanaee-T, Hadi, and Gama, Joao, "Event labeling combining ensemble detectors and background knowledge," Progress in Artificial Intelligence (2013): pp. 1-15, Springer Berlin Heidelberg, doi:10.1007/s13748-013-0040-3."

## 2.5 Tools

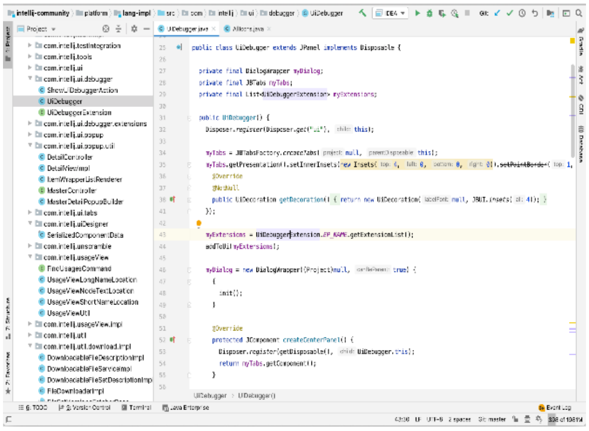
### 2.5.1 IntelliJ IDEA

To create advanced or simple IT programs that a programmer or organization needs, a code editor is necessary. One of the best editors is IntelliJ IDEA, a commercial development environment that can compile written code. This editor was created by the company "JetBrains" in Java. The first version was released in early 2001 and included tools for refactoring code.

Now IntelliJ IDEA offers many useful tips and keyboard shortcuts to make the programmer's work easier. IntelliJ IDEA supports 19 programming and automation languages, such as "Python," "Java," "Scala," and web development languages like "HTML" and "CSS." This allows developers to work in different languages in a single, consistent environment. IntelliJ IDEA also works with many other open source tools and environments, such as "GIT," "SVN," "CVS," "Apache Maven," "Apache Ant" and "JUnit," making project management and integration with other systems much easier. Also of interest to many IT fans is the dark interface theme of this editor, which reduces eye fatigue during prolonged work. IntelliJ IDEA is not only a tool for writing code, but also supports the programmer with features such as smart hints, code auto-completion, real-time error analysis and comprehensive debugging options. All this makes it a tool valued by both beginners and experienced programmers. The editor is also regularly updated, providing access to the latest technologies and best practices in programming. With an active user community and a broad knowledge base available online, support and skill development are at your fingertips. IntelliJ IDEA is an investment that pays dividends in increased productivity and code quality.



```
@app.route('/upload_files', methods=['GET', 'POST'])
def upload():

    driver_path = 'C:/IDEA/Project/diploma/webdrivers/chromedriver.exe'
    chrome_options = webdriver.ChromeOptions()
    # chrome_options.add_argument("--headless")

    path = 'C:/IDEA/Project/diploma/files/upload'
    if request.method == "POST":
        if request.files:
            if len(os.listdir(path)) != 0:
                for filename in os.listdir(path):
                    filepath = os.path.join(path, filename)
                    os.remove(filepath)

            upload_file = request.files["excelFile"]
            upload_filename = upload_file.filename
            delay = request.form.get("delay")

            if delay == "" or delay == " ":
                delay = 0.5
            float(delay)

            if upload_filename != "":
                message = ""

                upload_file.save(os.path.join(path, upload_filename))
                data_filename = upload_filename
                data_file_path = path + "/" + data_filename

                data_tab_headers = pd.read_excel(data_file_path, 'Data', header=None)
                data_tab_value = pd.read_excel(data_file_path, 'Data')
                reader = pd.read_excel(data_file_path, 'Data', header=None, [illegible])
                data_col_count = len(reader.columns)

                wb = load_workbook(filename=data_file_path)
                ws = wb['Data']
```



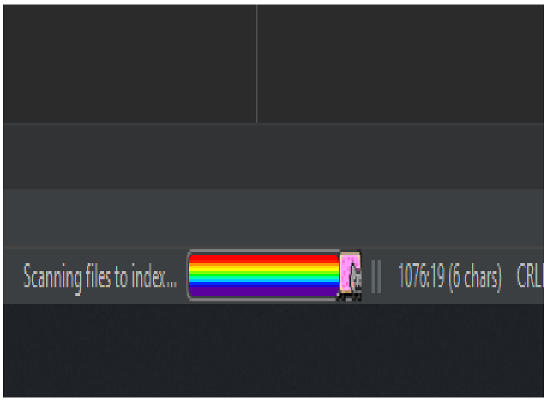
```
public class UiDebugger extends JPanel implements Disposable {

  private final DialogWrapper myDialog;
  private final JBTabs myTabs;
  private final List<UiDebuggerExtension> myExtensions;

  public UiDebugger() {
    Disposer.register(Disposer.get("ui"), child: this);

    myTabs = JBTabsFactory.createTabs( project: null, parentDisposable: this);
    myTabs.getPresentation().setInnerInsets(new Insets( top: 4, left: 0, bottom: 0, right: 0)).setPaintBorder( top: 1,
      @Override
      @NotNull
      public UiDecoration getDecoration() { return new UiDecoration( labelFont: null, JBUI.insets( all: 4)); }
    });

    myExtensions = UiDebuggerExtension.EP_NAME.getExtensionList();
    addToUi(myExtensions);

    myDialog = new DialogWrapper((Project)null, canBeParent: true) {
      {
        init();
      }

      @Override
      protected JComponent createCenterPanel() {
        Disposer.register(getDisposable(), child: UiDebugger.this);
        return myTabs.getComponent();
      }
```

Picture 16 Left side, IntelliJ dark theme Picture 17 Right side, IntelliJ default light theme

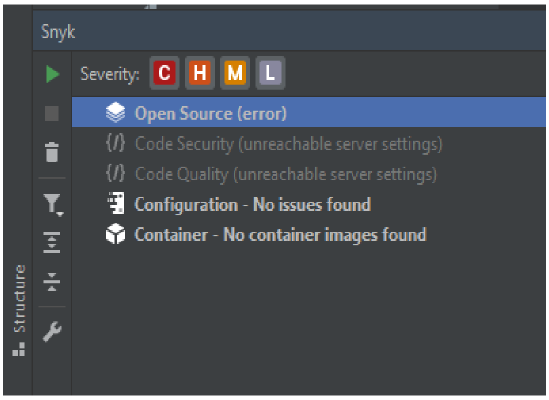
To extend the functionality of the basic version of IntelliJ IDEA, many plug-ins are available. One example is "Rainbow Brackets" by "izhangzhihao", which improves code readability by coloring overlapping round, square and bracket brackets. Another plug-in, "Snyk Security - Code, Open Source, Container, IaC Configurations," analyzes code written by a programmer and generates a report on application security. The reports are very clear, even for novice programmers, making it easier to understand potential problems. "Snyk" also offers possible solutions for detected security vulnerabilities. There are also recreational plug-ins that aim to make IntelliJ IDEA users feel better. One of them is the "Nyan Progress Bar" created by

"Dimitry Batkovich," which turns the compiler loading bar into a rainbow tail of the programming community's cult hero "Nyan Cat." With such additions, working with the editor becomes not only efficient, but also enjoyable. Expanding IntelliJ IDEA's capabilities with such plug-ins makes the editor even more versatile and customizable. Whether it's improving code readability, enhancing application security or introducing an element of fun, the available plug-ins can make developers' daily work much easier and more enjoyable.

(Jetbrains, 2024)



Scanning files to index... 1076:19 (6 chars) CRL



Snyk

Severity: C H M L

Open Source (error)

Code Security (unreachable server settings)

Code Quality (unreachable server settings)

Configuration - No issues found

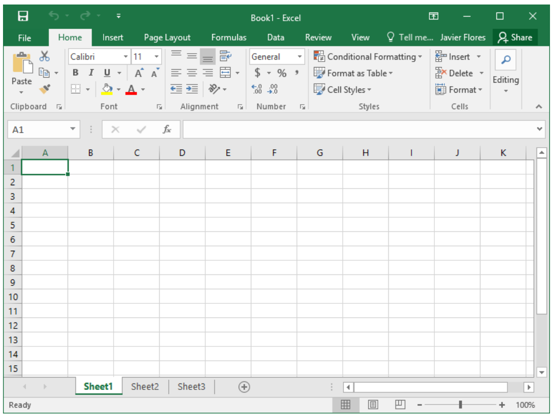
Container - No container images found

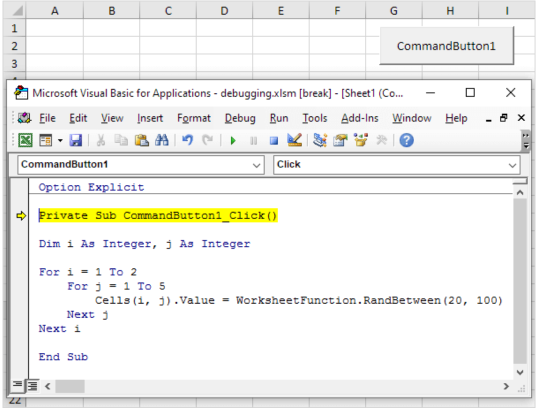
**Picture 18 Left side, Nyan Progress Bar Picture 19 Right side, Snyk add-in**

### 2.5.2 Excel

Microsoft Excel is a popular spreadsheet available for Windows, macOS, Android and iOS. It allows users to perform various mathematical calculations and present data using charts and pivot tables. With the Macros feature, you can also automate repetitive tasks. Macros are created using the Visual Basic for Applications (VBA) language. Each new version of Excel introduces a number of new features and improvements that make the program more user-friendly. In addition to the standard features, there are also add-ons that extend the program's capabilities. These add-ons, often developed by independent developers, offer additional tools and features that are not included in the official version of the program. With these extensions, Microsoft Excel becomes an even more flexible and powerful tool, ideal for advanced data analysis and automation of various processes, significantly increasing its value in both business and everyday applications.

(Microsoft, 2024)





```
Option Explicit

Private Sub CommandButton1_Click()

Dim i As Integer, j As Integer

For i = 1 To 2
    For j = 1 To 5
        Cells(i, j).Value = WorksheetFunction.RandBetween(20, 100)
    Next j
Next i

End Sub
```

**Picture 20 Left side, Excel Worksheet Picture 21 Right side, Visual Basic for Applications**

### 2.5.3 Python

Python is a programming language that offers many possibilities thanks to its numerous tools and libraries. Its main advantage is the clarity and simplicity of its code, which makes it easy to learn and use. Python allows you to write code in a clear and concise manner. In Python, you don't have to declare variable types, such as integer or string, which makes it easier to write code. Due to its popularity, Python runs on many operating systems. Its development is led by the "Python Software Foundation," a non-profit organization. The standard version of Python, "CPython" is written in C. There are other versions, such as JPython, written in Java, and IronPython for .NET. Python was created in the early 1990s by Guido van Rossum. The language was created as a successor to ABC at the Center for Mathematics and Computer Science in Amsterdam (CWI - Centrum voor Wiskunde en Informatica). An interesting fact is that the name Python, which the author named this programming language, is not derived from the name of a species of snake. When the author chose the name for this programming language, he was a big fan of a comedy series aired by British television on "BBC" called "Monty Python's Flying Circus." Python is used in many fields such as data analysis, artificial intelligence, web development, task automation and game development. Frameworks such as Django and Flask help create websites, while libraries such as NumPy, pandas and TensorFlow are used in data analysis and machine learning. With a large community and many tools available, Python is an ideal choice for both beginners and experienced programmers. It is a language that can be easily adapted to different projects and tasks, making it a very versatile tool in programming.

(Python Software Foundation, 2024)

Picture 22 Left Side, CWI building Picture 23 Right side, Guido van Rossum

### 2.5.4 **Pandas**

Pandas is a library for the Python programming language, created to handle and analyze a variety of data, especially the numerical ones used in data analysis tasks. It was created and released under a "BSD" license in 2008 by Wes McKinney, who was then working at "AQR Capital Management." The motivation for creating this library stemmed from the need for a powerful tool for effective financial data analysis. The name "Pandas" comes from the term "panel data," which refers to datasets containing observations from different time periods in econometrics.

(Pandas, 2024)

Picture 24 Wes McKinney

### 2.5.5 NumPy

NumPy is a very popular library, used by programmers writing their code in Python programming language. The library is used for numerical calculations, and is invaluable when complex mathematical calculations involving large data sets are required. It performs calculations much faster than standard methods available in the raw Python programming language. This is due to the fact that the NumPy library is written in C. The entire NumPy library is developed as an open-source project, which in practice means that any user can contribute to the development of this powerful programming tool. Unfortunately, this does not mean that the author's ideas and written functionality will be added to the official version of the library, because the development of NumPy is managed by a group of lead developers, who diligently and carefully make key decisions on the direction of further development of the project. On the technical side, the NumPy library allows you to work with multidimensional arrays. Such arrays are called "ndarray". When you expand this name, you get the expression "n-dimensional array". Moreover, using the NumPy library it is possible to perform mathematical operations on entire arrays simultaneously, which significantly reduces the time of performing mathematical operations than when performing them on individual elements in the array. The history of this library goes back to the 1990s, while it functioned as another library called "Numeric." It wasn't until 2005 that American computer scientist and businessman Travis Oliphant decided to combine the functionalities of the "Numeric" library with the "Numarray" library to obtain the "NumPy" library we know today. NumPy is a library that is the basis for many other libraries created and designed for data analysis, as well as machine learning.

(NumPy, 2024)

**Picture 25 Travis Oliphant**

### 2.5.6 **Matplotlib**

Matplotlib is another library associated with machine learning, intended for use by programmers who use the Python programming language to write their code. The Matplotlib library is used to create graphs, as well as data visualization in the broadest sense. It allows you to generate graphical projections derived from a wide range of a group of graphics for data visualization. Starting from simple line charts, and ending with complex and advanced diagrams representing a specified range of data. It is also worth mentioning that Matplotlib allows you to create interactive charts, which allows you to dynamically modify the graphs. On the other hand, charts can be easily customized according to the user's needs in terms of colors, fonts and other elements contained in the generated graphical projections. Very often it appears in programming projects in tandem with such libraries as "pandas" or "NumPy". The popularity of this library has been with it from the beginning of its existence, when the author, John D. Hunter in 2003 decided to create a set of tools available to users of the Python programming language, similar in function to the graphing tools available in "MATLAB". John D. Hunter was a neuroscientist who needed a tool that met his requirements to be able to visualize data while constructing his research paper. Matplotlib, like most libraries available for use in the Python programming language, belongs to "open-source" projects, where users can report bugs, propose new features and participate in discussions about the future of the project, which is the "Matplotlib" library. After the death of John D. Hunter in 2012, the project is managed by a group of core developers who continue his work and take care of the library's development.

(Matplotlib, 2024)

**Picture 26 John D. Hunter**

### 2.5.7 Seaborn

Seaborn is a library available for use in the Python programming language, which is used to create statistical data visualizations. It is built on Matplotlib and provides a higher-level interface for creating attractive and informative charts. The Seaborn library is used to create visualizations that make it easier to understand statistical relationships between data. Seaborn offers simple ways to generate complex charts, such as correlation charts, distribution charts, regression charts, pair charts and more. In the development community, this library is known for its ease of use, offering an intuitive interface that makes creating advanced and complex charts simple and fast. Moreover, the library works seamlessly with the "pandas" library and the typical "Pandas DataFrame". This is crucial, because this integrity allows for the easy creation of charts directly from structured data, reprocessed by the tools available in the "pandas" library. Because of this, Seaborn is gaining in popularity and frequency of use in programming projects. Of the other features that deserve mentioning of this library are the built-in tools for using statistics in visualizations, such as box plots or heat maps. The author of this set of tools is neuroscientist, statistician Michael Waskom. He created the Seaborn library in 2012 to facilitate the creation of statistical graphs in the Python programming language. To this day, Michael Waskom plays a key role in the development of the Seaborn project, but other developers and users also contribute to its development by reporting bugs, proposing fixes and new features on an open-source project basis.

(Seaborn, 2024)

**Picture 27 Michael Waskom**

### 2.5.8 Scikit-learn

Scikit-learn, often stylized as "sklearn," is a popular library in the Python programming language used for machine learning. It is easy to use and offers a wide set of tools for data analysis and modeling. This library is very often chosen as a tool for building and evaluating machine learning models. Scikit-learn supports various techniques related to machine learning, and it is worth mentioning classification, regression, clustering, dimensionality reduction, as well as data preprocessing tools. Comprehensiveness is definitely a term that can be used when discussing the Scikit-learn library, as it offers a very wide range of machine learning algorithms. What's more, it interfaces perfectly with the Python programming language environment, as well as libraries dedicated to this language, such as NumPy, SciPy, Pandas and Matplotlib. The author of this library is David Cournapeu, it was created as a "Google Summer of Code" project in 2007.

(Scikit-learn, 2024)

# 3 Analysis process assumptions

The first step in carrying out the analysis was to find a suitable dataset that would contain a large amount of data and would be related to the topic on the basis of which it would be possible to carry out an analysis leading to very interesting conclusions. I spent a lot of time to find such a dataset, one of the main aspects determining my choice was to what extent the subject matter in the form of data would be interesting to me. I looked through dozens of datasets available on the "UC Irvine" portal. Ultimately, my attention was focused on a dataset presenting information on urban bicycle rentals. I have repeatedly wondered how such urban bicycle rental systems work, among other things whether it's a profitable business for the companies running such services from the perspective of interest in using such bicycle rental by users. It was these questions that prompted me to decide to analyze the "Bike Sharing" dataset. I then carefully read through the descriptions of the attributes in the dataset to make initial assumptions. The assumptions included deciding which of the attributes in the "Bike Sharing" dataset, when put together, would lead to interesting conclusions in order to maximize the statistical potential found in this dataset. Having determined the goals I would like to achieve by analyzing the data, I proceeded to select the appropriate tools for specific tasks, supporting me in carrying out the analysis. The choice fell on the program "Excel" by "Microsoft" from the "Office" package, because it offers many forms for processing data and presenting them in graphical form, moreover, this program supports the ability to read multiple file extensions. In my case, the dataset "Bike Sharing" is saved using the file extension "CSV", which "Excel" handles without any problems. I decided that in addition to analyzing the juxtaposition of specific attributes with each other, an interesting part of the analysis would be to compile information on how important specific attributes are in terms of their impact on the number of rented bicycles over the entire time range presented in the "Bike Sharig" dataset. To accomplish this task, I relied on training the model involving machine learning. In order to properly train the machine learning model and obtain the results I was interested in, based on the data structure present in the "Bike Sharing" dataset, I decided to use linear regression. Linear regression is one of the simplest methods for modeling relationships between variables. Regression coefficients directly show how a change of one unit in a trait affects the outcome. This makes it easy to understand the effect of individual characteristics on the number of rented bicycles. In many cases, the relationship between variables and outcome can be well approximated by a linear function. In the context of bicycle rentals, characteristics such as temperature, day of the week or wind can have a linear effect on the number of bicycles rented.

Using linear regression, the significance of individual characteristics can be easily assessed. The regression coefficients and their values allow you to identify which variables have the greatest impact on the outcome. Linear regression is a good choice for training a bicycle rental analysis model because it is simple, easy to interpret, computationally efficient and works well with a large number of observations. In addition, it allows for easy assessment of the significance of features and provides a solid foundation for possible extension to more advanced models if it were necessary. Moving on, before starting to draw conclusions, it is necessary to purge the dataset of information that will not necessarily be useful during the analysis or that may even have a negative impact on the final results of the analysis being conducted. When the analysis is carried out using "Excel", unnecessary attributes do not pose a problem, only the empty values contained in them, the so-called "NULLS", because when writing statistical forms, it is enough not to include those attributes in the calculations that do not have much meaning or negatively affect the analysis carried out. On the other hand, at the time of training a machine learning model that scans the entire data set, the attributes that negatively affect the result of the training should be cut from the area of the analyzed data set, and in the next stage of the analysis. Such attributes are "instant", "dteday", "casual" and "registered". The attribute "instant", was not qualified due to the fact that it only represents an iterative entry number in the "Bike Sharing" dataset. The item "dteday" was excluded because it does not have much meaning for analysis, but only serves to represent other attributes on its basis, for example, the attribute "season" representing the time of year. The situation is similar with the attributes "casual" and "registered" because the sum of their values is the target attribute in the automated analysis, we are talking about "cnt" representing the exact number of rented bicycles on a particular day. On the other hand, the "casual" and "registered" attributes have their uses in manual analysis.

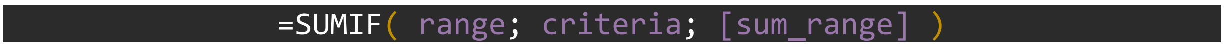
# 4 Implementation

## 4.1 Implementation of manual analysis

The analysis process performed manually leads through eight scenarios. Each of the statements performed shows a different factor affecting the number of bicycles rented in the scenario's assumed time period.

### 4.1.1 Seasonal analysis implementation

Seasonal analysis is a compilation of four outcome data. Each value represents a specific season among spring, summer, autumn and winter. The data is calculated using a statistical formula available in "Excel" called "SUMIF". As the name suggests, the "SUMIF" formula allows you to sum specific values at the moment if a specific condition, defined and written in the formula, is met. The structure of the entire formula is as follows, shown below.

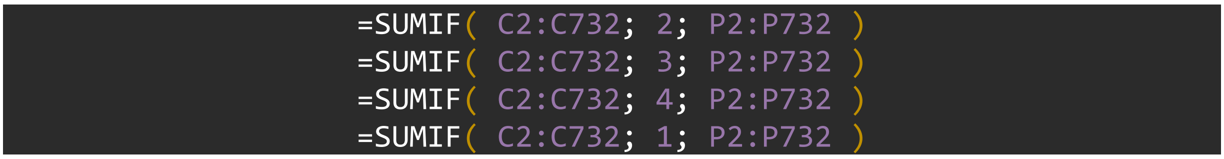


```
=SUMIF( range; criteria; [sum_range] )
```

**Figure 2 SUMIF formula structure**

This structure represents a three-element formula. The first element is the range of cells located in a specific sheet submitted to the analysis performed by the "SUMIF" formula functionality. The second element is the criteria that should be met in the defined range of data in the first element of this formula. The last element is the range of cells that should be summed when the condition defined in the second item of the formula, located under the term "criteria", is met.

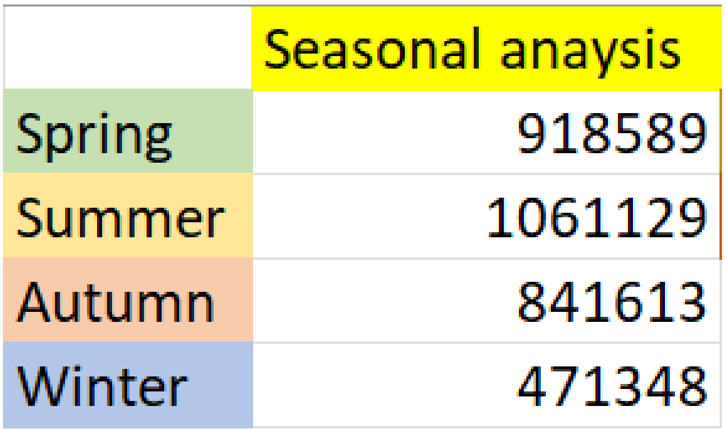
In the case of seasonal analysis, the specific values found in the "SUMIF" formula are those shown below.



```
=SUMIF( C2:C732; 2; P2:P732 )
=SUMIF( C2:C732; 3; P2:P732 )
=SUMIF( C2:C732; 4; P2:P732 )
=SUMIF( C2:C732; 1; P2:P732 )
```

**Figure 3 Seasonal analysis formulas**

The first item is a range of cells written using cell references representing the column letters and row numbers of the season attribute. The element in the second position is the criteria that determines the selection of appropriate values in the column containing data on seasons. Number 1 represents winter, under number 2 records collected in spring are described, 3 is summer, and number 4 is assigned to fall entries. The last element referring to the summed values when the declared condition located under the second element of the "SUMIF" formula is the range of cells marked with the "cnt" attribute, which corresponds to the number of rented bicycles of a particular day.



|  | Seasonal anaysis |
|---|---|
| Spring | 918589 |
| Summer | 1061129 |
| Autumn | 841613 |
| Winter | 471348 |

Figure 4 Output values of seasonal analysis

The numerical results of the analysis conducted show the summer season as the season when the most urban bicycles are rented over the two years. In contrast, the season when rented bicycles are the least is the winter season.

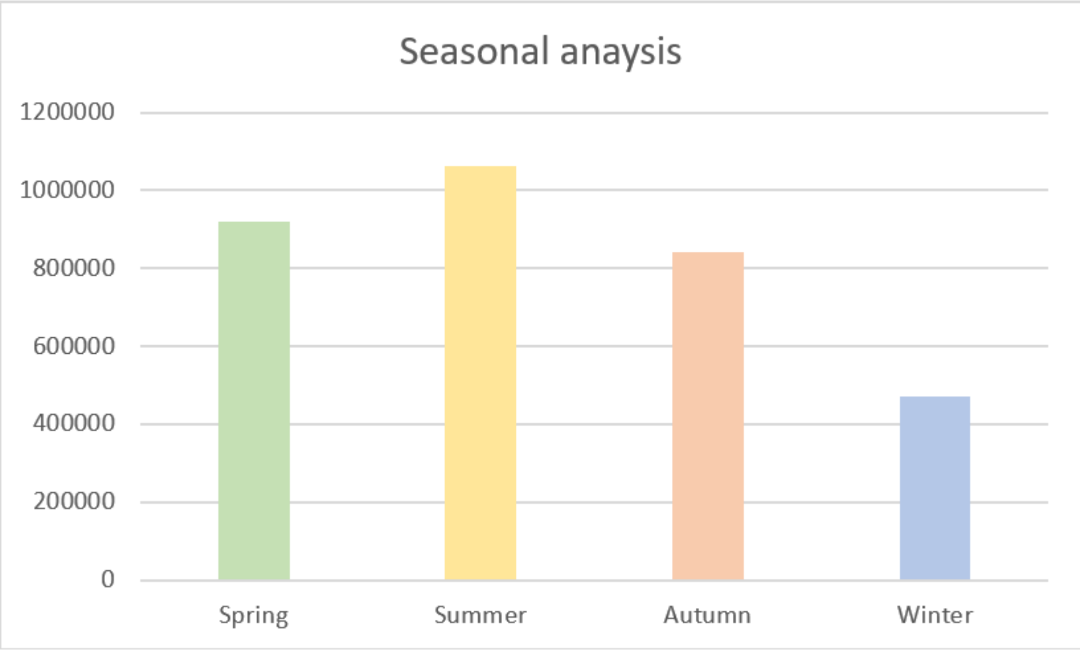
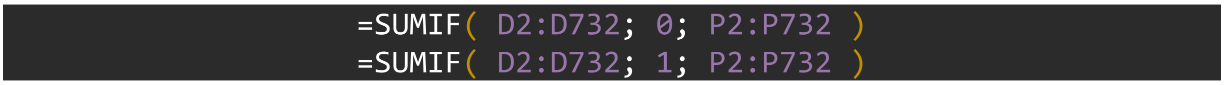


Figure 5 Seasonal analysis chart

The graph shown is an easier form of representation of the results of the analysis for the analyzer. Thanks to this bar chart, we can see an almost uniform tendency to wave the results between the four seasons based on data collected over two years. In addition to the maximum and minimum values belonging to the summer and winter seasons, we can see an increased number of rented bicycles for the spring period in relation to the fall season. Analyzing the entire graph further, taking into account all the items on the horizontal axis, we can conclude that the number of rented bicycles can be influenced by the different temperature accompanying each of the presented seasons. Because, as a rule, the spring season is a warmer season than the autumn season. On the other hand, the extreme temperature opposites are the summer and winter seasons, which represent the first and last ranked seasons in terms of the number of rented bicycles.

### 4.1.2 Annual analysis implementation

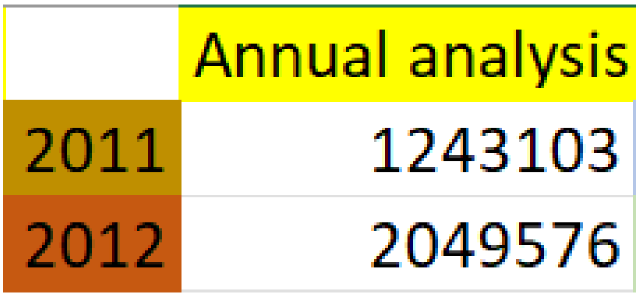
As in the previous scenario, a formula belonging to the standard and basic package of formulas in "Excel" was used to conduct the analysis.



```
=SUMIF( D2:D732; 0; P2:P732 )
=SUMIF( D2:D732; 1; P2:P732 )
```

Figure 6 Annual analysis formulas

The "SUMIF" formula in this case made it possible to determine the number of urban bicycles rented by the years in which the rentals were made. The first element of the "SUMIF" formula is the range of cells that were verified. This range represents the "yr" attribute in the "Bike sharing" dataset. The second element of the formula is the criteria, which was defined to segregate the results based on the values found in the set of cells declared in the first element of the "SUMIF" formula. The values of the declared criteria represent the years in effect in the dataset and regarding the year in which the bikes were rented. The number 0 represents 2011, while the number 1 refers to 2012.



| | Annual analysis |
|---|---|
| 2011 | 1243103 |
| 2012 | 2049576 |

**Figure 7 Output values of annual analysis**

The "SUMIF" formulas presented above showed a numerical summary of the annual analysis. The year 2012 definitely exceeds the number of rented city bicycles in relation to the previous year, namely in relation to 2011.

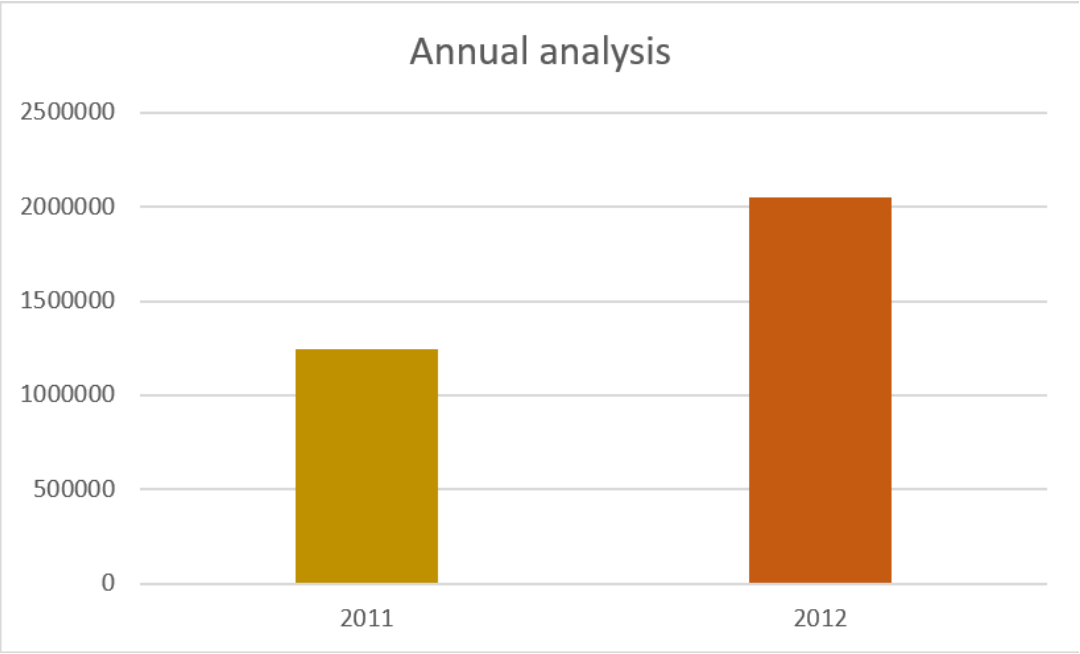
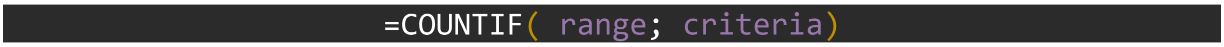


**Figure 8 Annual analysis chart**

The chart presented here illustrates in a much better way how much the results of rented bicycles in 2012 exceed the number of rentals from 2011. The increase in the number of rented bicycles over the two years increased by 54 percentage points. Such a result may suggest increased interest in the bicycle rental program, or even an observable increase in the public's habit of this mode of transportation. Many companies are even using forms of rewarding users for recommending their product to subsequent users who have not used the company's services before.

### 4.1.3 Impact of working days and holidays analysis implementation

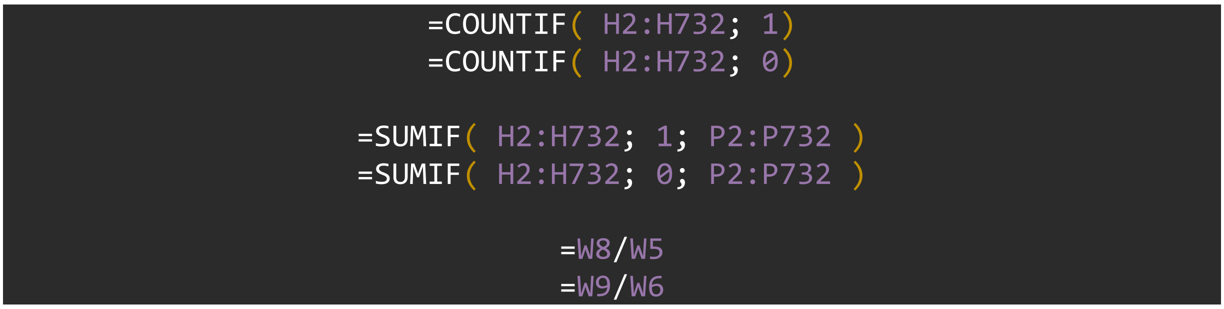
The analysis of this scenario is based on two output values, while the process of calculating them is more complex, since the analysis takes into account the fact that there are many more weekdays during the year than holidays taking into account also the days that are holidays in force in the country of origin of the data contained in the "Bike Sharing" dataset. To account for this disparity, I used the "COUNTIF" formula, which is also included in the package of standard and basic formulas in the "Excel" spreadsheet program.



```
=COUNTIF( range; criteria)
```

Figure 9 COUNTIF formula structure

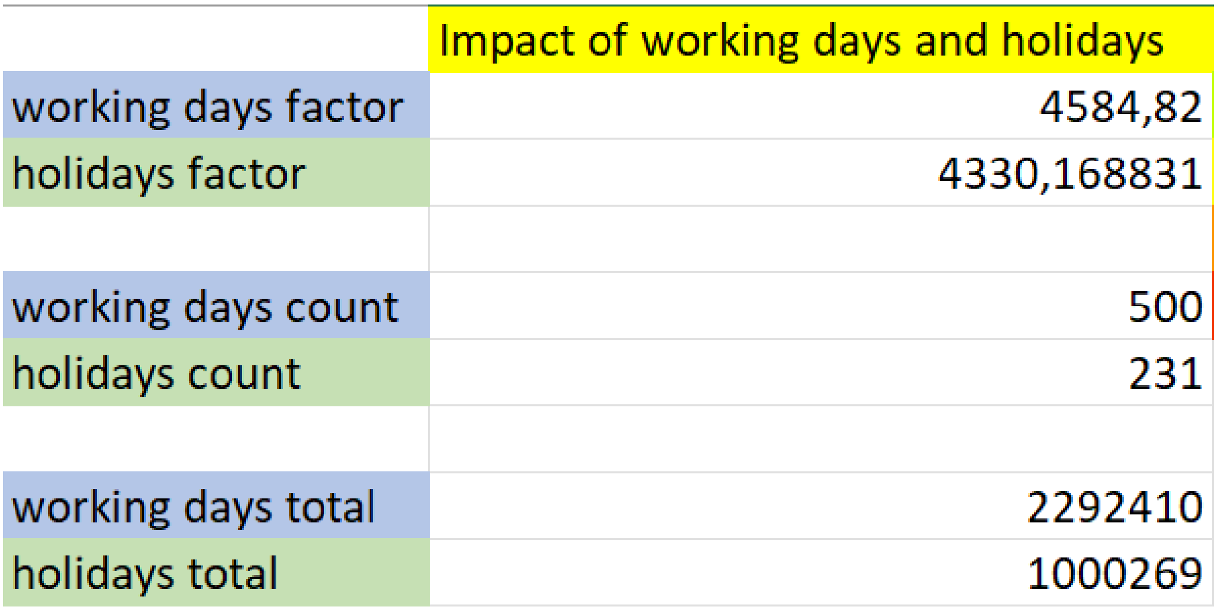
The "COUNTIF" formula consists of two parameters. The first is the range of cells containing data, the number of which is then counted based on the defined condition. The second parameter is the criteria, which at the same time is the condition required for a particular entry to be added to the pool of entries that meet the requirements against the cell range declared in the first parameter.



```
=COUNTIF( H2:H732; 1)
=COUNTIF( H2:H732; 0)

=SUMIF( H2:H732; 1; P2:P732 )
=SUMIF( H2:H732; 0; P2:P732 )

=W8/W5
=W9/W6
```

Figure 10 Impact of working days and holidays analysis formulas

The statistical formulas presented above are divided into three segments, each calculating a different but equally key value in the context of the final result that was analyzed. The first segment, written using the "COUNTIF" formula, is the segment responsible for calculating the exact number of days of a particular type occurring, among working days and non-working days. The first segment of this formula is a range of cells named with the attribute "workingday" containing information on whether a particular entry in the dataset represents a working day denoted by the number 1 or a non-working day denoted by the number 0 in the dataset. The next segment of this formula is the criteria by which it is possible to separate the number of working days from non-working days. The second segment of the formulas shown above is the segment containing "SUMIF" formulas. This segment allows you to count the number of rented bicycles in a set of cells containing information about the type of day among employee days and days off, defined in the first element of the formulas located in the "SUMIF" segment. As in the previous segment, the criteria are values of 0 or 1, which represent the type of day among employee days and days off. The element that contains the range of cells whose values are summed are the cells marked with the "cnt" attribute, representing the exact number of rented city bikes for each of the entries in the dataset. The last segment of formulas are formulas showing the final result. This is the basic type of formula that usually uses a mathematical operation, which in this case is division. The division used calculates how many average daily urban bicycles are rented for work days and how many average urban bicycles are rented for non-work days.



| | Impact of working days and holidays |
|---|---:|
| working days factor | 4584,82 |
| holidays factor | 4330,168831 |
| | |
| working days count | 500 |
| holidays count | 231 |
| | |
| working days total | 2292410 |
| holidays total | 1000269 |

Figure 11 Output values of impact of working days and holidays analysis

The numerical values presented above show how important it is to divide the sum of all rented bicycles among employee days and holidays by the number of days occurring for the specific category analyzed, since for fields described as "total," the numerical values differ by more than double, while when calculating the values described as "factor," the difference decreases significantly.

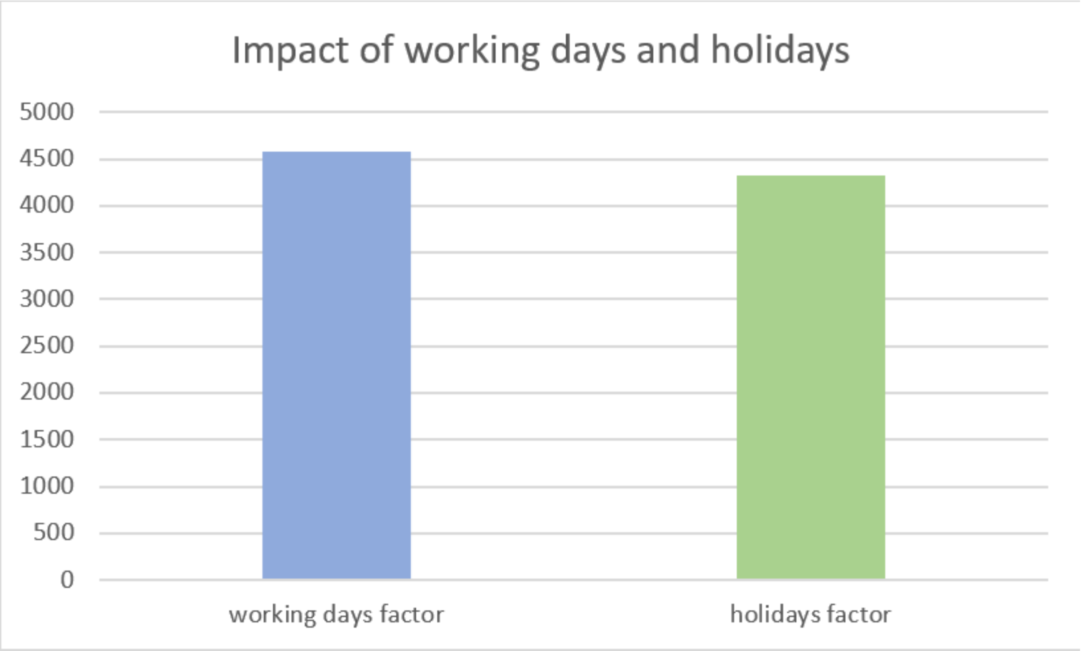
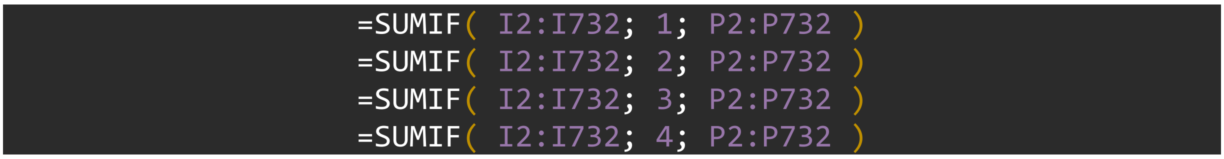


**Figure 12 Impact of working days and holidays analysis chart**

In the chart shown, one can see literally little difference from the average number of urban bicycle rentals for the two cases analyzed from among work days and non-work days. The conclusion drawn from this analysis is that the type of day type among working days and non-working days has a negligible effect on the number of urban bicycle rentals. However, this insignificant difference may be due to the fact that a certain group of minority users may spend their holidays with their families rather than using a means of transportation such as city bicycles.

### 4.1.4 Weather influence analysis implementation

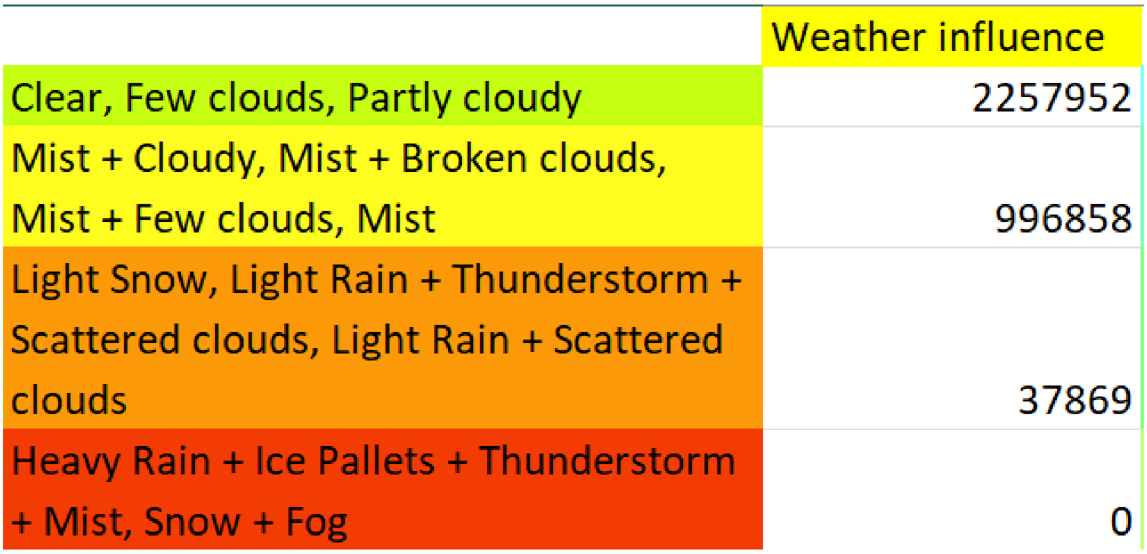
The effect of weather on the number of city bicycles rented over two years is a very interesting factor depicting the extent to which the public responds to changing weather.



```
=SUMIF( I2:I732; 1; P2:P732 )
=SUMIF( I2:I732; 2; P2:P732 )
=SUMIF( I2:I732; 3; P2:P732 )
=SUMIF( I2:I732; 4; P2:P732 )
```

**Figure 13 Weather influence analysis formulas**

To carry out this analysis, I used the "SUMIF" formula, which in the first element of its structure defines a range of cells containing data described by the "weathersit" attribute. The data are presented in numerical form, with numbers in the range from 1 to 4. The number 1 represents days in which the weather could be described as days with clear skies, a few clouds, partial cloud cover. A value of 2 refers to a combination of the presence of fog and cloud cover, fog and considerable cloud cover, fog and little cloud cover, and fog alone. Entries marked with the number 3 are days with light snow, light rain, thunderstorms and scattered clouds, and light rain and scattered clouds. The last category, marked by the number 4, is heavy rain, icing, thunderstorm with fog, snow with fog. The second segment of the "SUMIF" formula is the criteria by which it is possible to separate the entries in the "Bike Sharing" dataset into categories for the weather occurring on a given day. The last segment is a range of cells containing data on the number of urban bicycles rented, this range is marked with the "cnt" attribute.



| | Weather influence |
|---|---|
| Clear, Few clouds, Partly cloudy | 2257952 |
| Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist | 996858 |
| Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds | 37869 |
| Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog | 0 |

**Figure 14 Output values of  weather influence analysis**

The above summary of numerical results, calculated using "SUMIF" formulas, shows a downward trend in the number of rented bicycles as the weather worsens. The key and striking element of these numerical results is the number of rented bicycles for the worst possible weather conditions, as the number of rented city bicycles over two years is all of 0 rented bicycles.

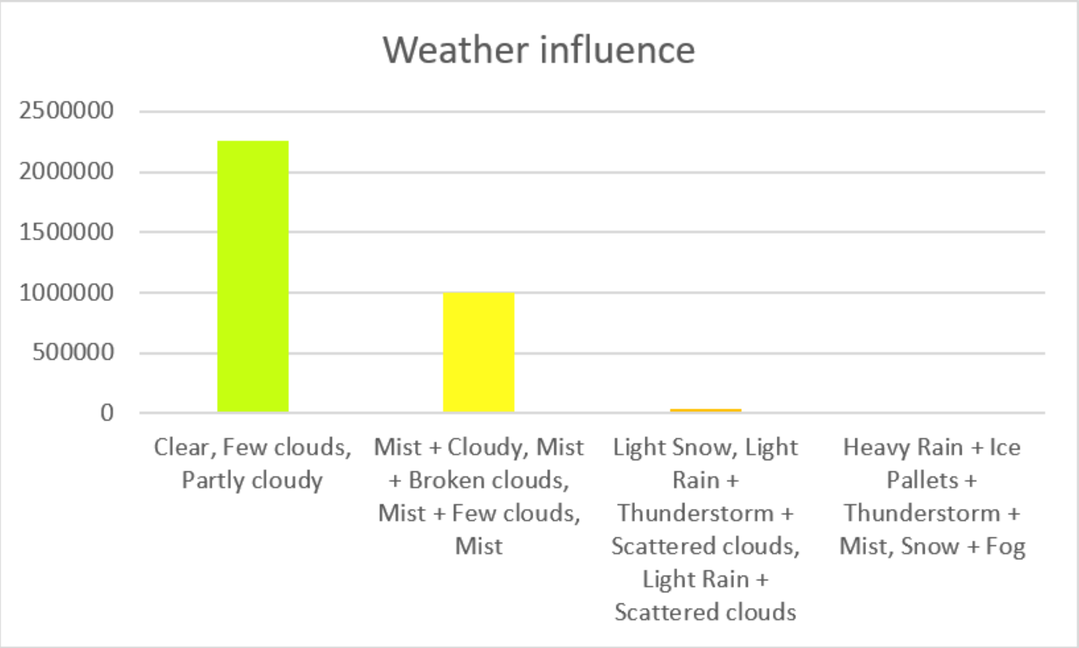
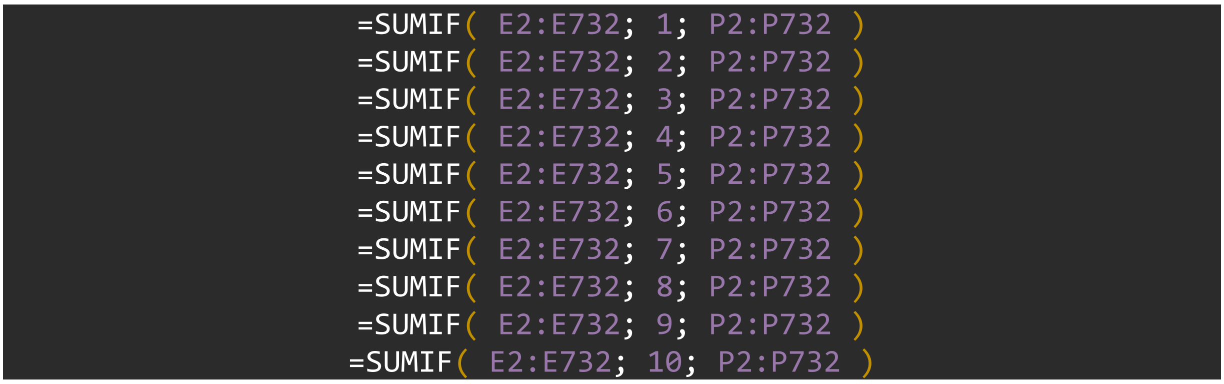


**Figure 15 Weather influence analysis chart**

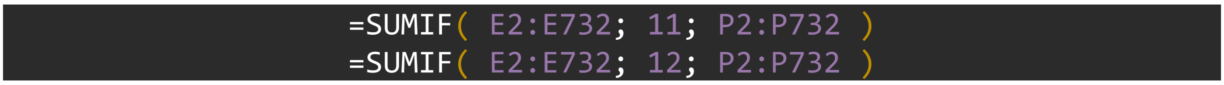
The graph shown above illustrates very well the decline in interest in urban bicycle rental with worsening weather described on a four-degree scale. The result of such analysis realizes in the fact that deterioration of weather by one degree based on the four-degree scale describing weather results in a decrease in interest in renting city bicycles by up to 50%. On the other hand, on days outside the first and second degree on the four-degree scale describing the weather, it is not worth counting on any cases of urban bicycle rental.

### 4.1.5 Monthly analysis implementation

The monthly analysis shows twelve items on an annual basis. With the help of such an analysis, we are able to assess which months are most profitable, at the same time as information on when interest in renting city bikes is highest. To conduct this analysis, I used the "SUMIF" formula.



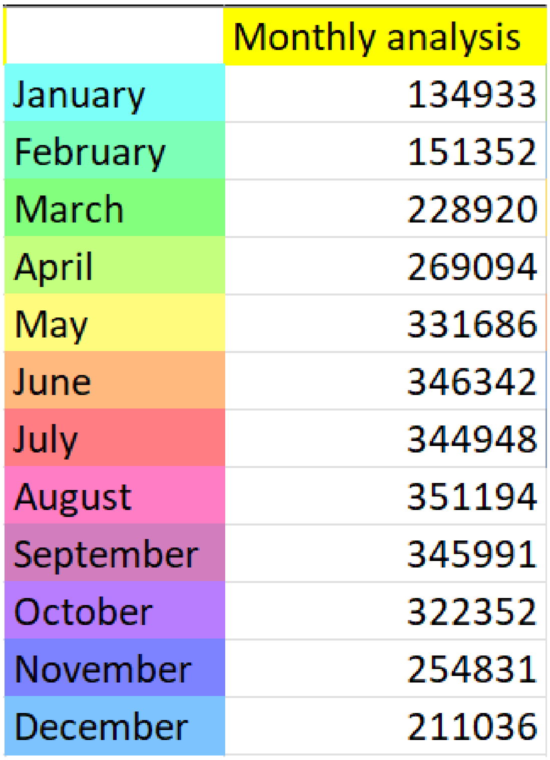
```
=SUMIF( E2:E732; 1; P2:P732 )
=SUMIF( E2:E732; 2; P2:P732 )
=SUMIF( E2:E732; 3; P2:P732 )
=SUMIF( E2:E732; 4; P2:P732 )
=SUMIF( E2:E732; 5; P2:P732 )
=SUMIF( E2:E732; 6; P2:P732 )
=SUMIF( E2:E732; 7; P2:P732 )
=SUMIF( E2:E732; 8; P2:P732 )
=SUMIF( E2:E732; 9; P2:P732 )
=SUMIF( E2:E732; 10; P2:P732 )
```



```
=SUMIF( E2:E732; 11; P2:P732 )
=SUMIF( E2:E732; 12; P2:P732 )
```

Figure 16 Monthly analysis formulas

The formulas shown above are three-segmented, where the first segment represents the range of cells subjected to selection, the condition of which is the value contained in the second segment. The range of these cells are the values that have monthly breakdown information using numerical identification, this range is marked with the attribute "mnth". The numbers represent a specific calendar month. Segment number two defines a condition based on the set of cells from segment one, which the formula is to follow when summing values from the range of cells defined in segment three. Segment three is the cells containing the values for the number of rented bicycles. These cells are marked with the "cnt" attribute.



| | Monthly analysis |
|---|---|
| January | 134933 |
| February | 151352 |
| March | 228920 |
| April | 269094 |
| May | 331686 |
| June | 346342 |
| July | 344948 |
| August | 351194 |
| September | 345991 |
| October | 322352 |
| November | 254831 |
| December | 211036 |

Figure 17 Output values of Monthly analysis

The resulting figures shown using the "SUMIF" formula represent the exact numbers of rented bicycles over the two years included in the "Bike Sharing" dataset analyzed, broken down by twelve months. Analyzing these figures, we are able to see that the smallest values occur in months associated with reduced air temperature.

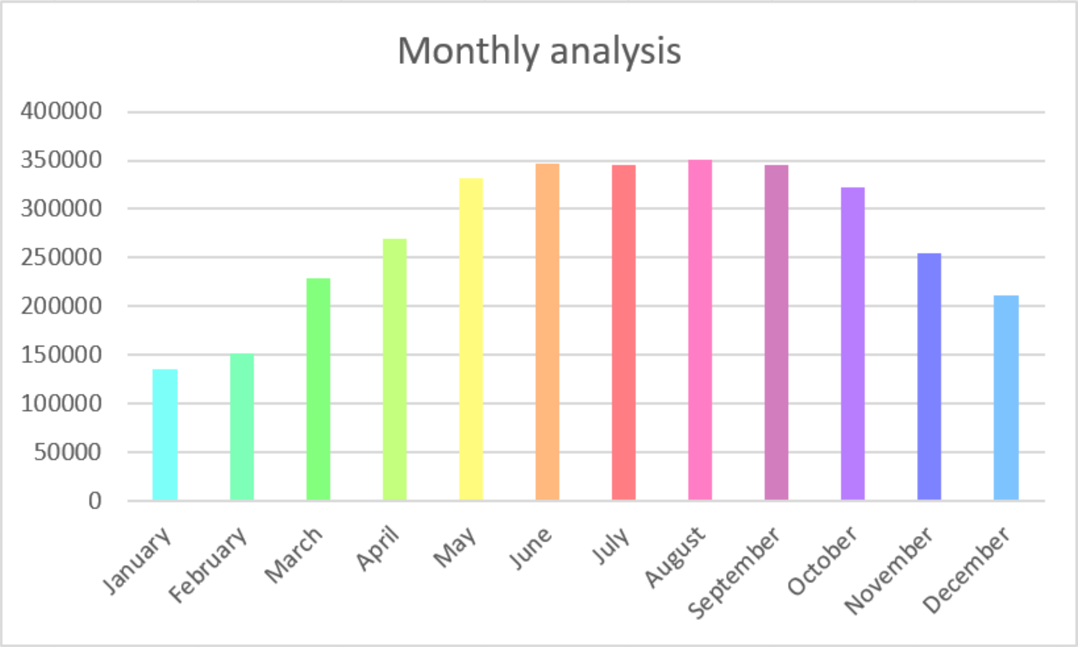


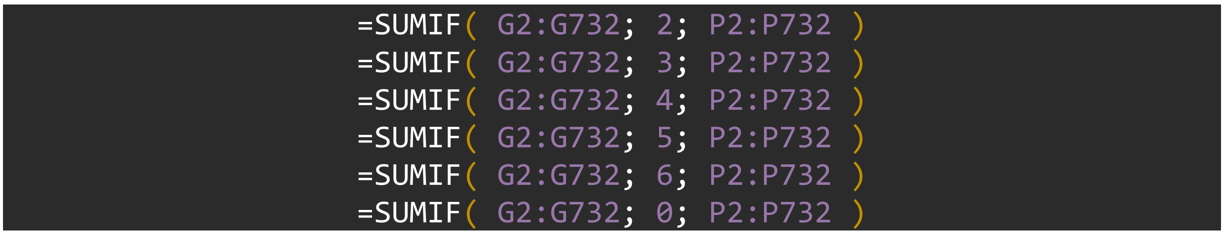
**Figure 18 Monthly analysis chart**

The graph of monthly analysis illustrates the extent to which there are differences in the number of rented urban bicycles divided into twelve months over the analyzed two years as seen in the "Bike Sharing" dataset. Analyzing this graph, it can be concluded that the months in which there is an increased interest in renting city bicycles fall in the middle part of the year. This may be due to the increased air temperature occurring during these months, which makes potential users much more inclined to an active lifestyle and use city bicycles as their primary means of transportation. An additional coexisting aspect that may influence the increased number of urban bicycle rentals in the middle part of the year is also the holiday season, when elementary and high school students, as well as students, have more free time due to the lack of current teaching activities during this period may positively influence active leisure activities such as urban bicycle travel.

### 4.1.6 Implementation of analysis by days of the week

An analysis that takes into account the days of the week as determinants of the number of urban bicycle rentals is a very interesting statement. Such an analysis can show on which days of the week users are most likely to rent bicycles. To carry out this analysis, I used a formula found in the basic set of formulas available in "Excel". The "SUMIF" formula will work perfectly to perform this analysis.

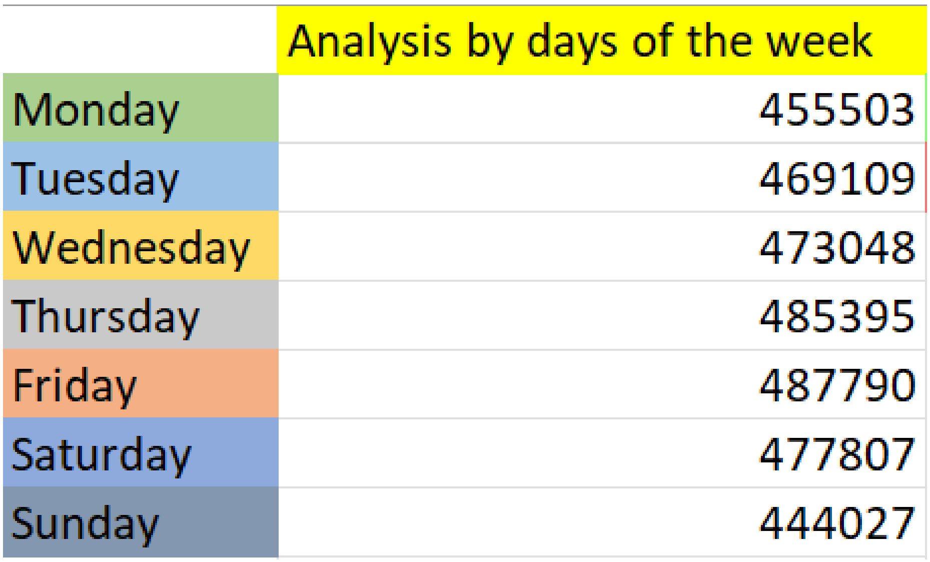
```
=SUMIF( G2:G732; 1; P2:P732 )
```



```
=SUMIF( G2:G732; 2; P2:P732 )
=SUMIF( G2:G732; 3; P2:P732 )
=SUMIF( G2:G732; 4; P2:P732 )
=SUMIF( G2:G732; 5; P2:P732 )
=SUMIF( G2:G732; 6; P2:P732 )
=SUMIF( G2:G732; 0; P2:P732 )
```

**Figure 19 Formulas for analysis by days of the week**

The first segment of the "SUMIF" formula structure contains a set of cells containing information about the days of the week on which measurements were taken. The days of the week were designated in numerical form, with numbers ranging from 0 to 6, where the number 1 represents Monday, 2 refers to Tuesday, 3 is Wednesday, 4 stands for Thursday, 5 represents days occurring on Friday, 6 represents Saturday, and the number 0 indicates days falling on Sundays. The numbering of these days of the week was used to define the criteria found in the second segment of the "SUMIF" formula structure, with the help of which the set of cells described by the "weekday" attribute, which was defined in the first segment of the "SUMIF" formula structure, was divided into each of the days of the week. The last segment of the "SUMIF" formula is a set of cells containing information about the exact number of rented bicycles broken down for each day in the range of data analyzed from among the two years appearing in the "Bike Sharing" dataset.



| | Analysis by days of the week |
|---|---|
| Monday | 455503 |
| Tuesday | 469109 |
| Wednesday | 473048 |
| Thursday | 485395 |
| Friday | 487790 |
| Saturday | 477807 |
| Sunday | 444027 |

**Figure 20 Output values of analysis by days of the week**

The resulting figures for analyzing the number of rented city bicycles by day of the week show that the numbers of rented bicycles by week vary slightly.

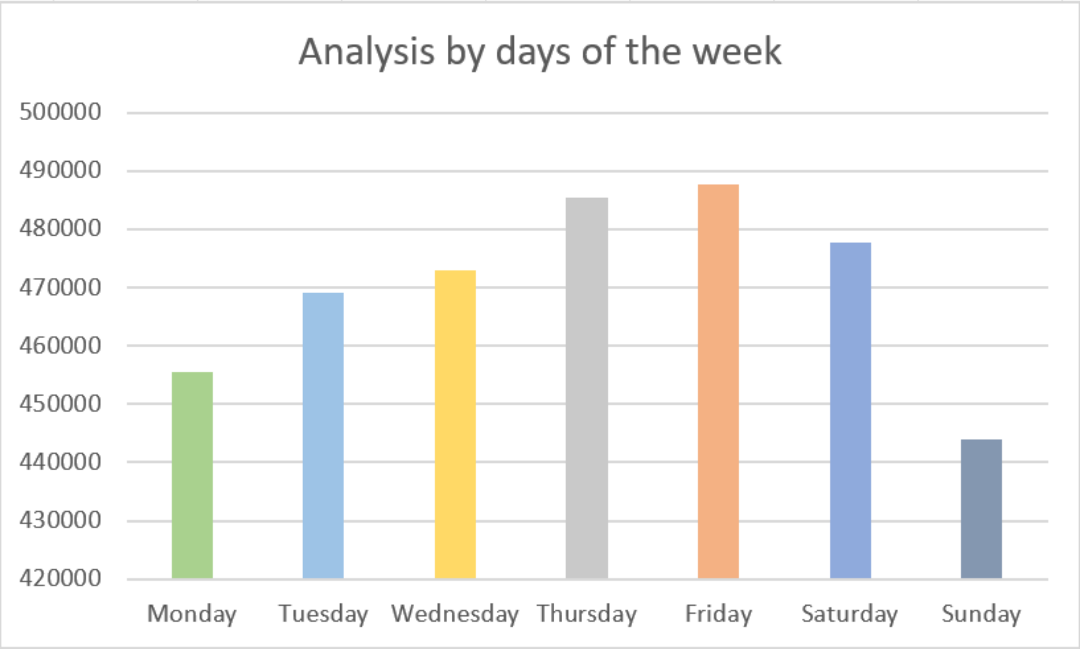
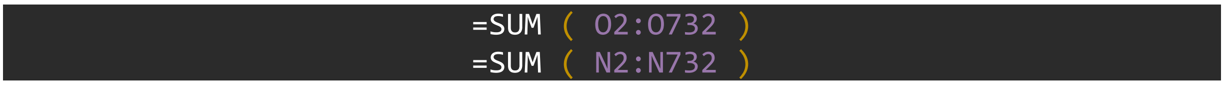


**Figure 21 Chart of analysis by days of the week**

Using a graph generated from the resulting figures, it is possible to see exactly which days of the week enjoy increased interest in renting city bicycles. The days with the top result of rented bicycles over the analyzed two years are Thursday and Friday. Such results may suggest that the increased interest in renting bicycles on these days are gatherings of friends, where it can be inconvenient to communicate with one's own vehicle such as a motorcycle or car, since statistically these are the days with the highest number of special events caused by the end of the work week and the start of weekend days off. From the second point of view, the day that received the lowest number of rented bikes was Sunday. Such a result may suggest that the reason for the underreported interest may be the religious character given to this day, resulting in an increased need to spend time with family around a common table, and eating a meal together, or even attending celebrations held at religious sites.

### 4.1.7 Comparison of registered and casual users analysis implementation

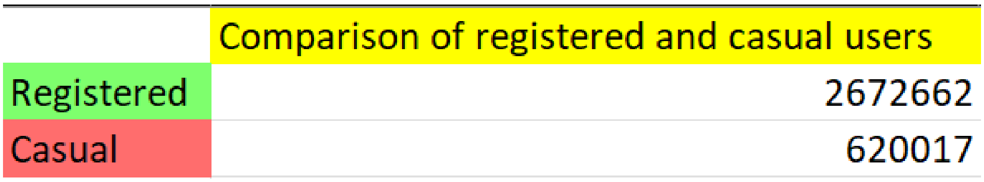
With the help of this analysis it is possible to assess what is the breakdown of the classification of users. The results of such an analysis are unlikely to have much direct impact on the number of rented bicycles. On the other hand, it will help illustrate what proportion of users are willing to create an account in an application or portal dedicated to a city bike rental program.



```
=SUM ( O2:O732 )
=SUM ( N2:N732 )
```

Figure 22 Comparison of registered and casual users analysis formulas

To perform this analysis, I used the "SUM" formula, which is ideal for this task. The choice of this formula was prompted by the data in the dataset required for this analysis. The data in the cell ranges described by the attributes "casual" and "registered" are the exact number of rented bikes for each group. The sum of these two values within a single entry, is the result of the number of rented city bikes for a specific entry in the "Bike Sharing" dataset.



| | Comparison of registered and casual users |
|---|---|
| Registered | 2672662 |
| Casual | 620017 |

Figure 23 Output values of comparison of registered and casual users analysis

The output figures show the exact number of bicycles rented by each of the analyzed user groups from among registered users and those who do not have registration with the application or portal that operates the urban bike rental program. Based on these output figures, we are able to see the dominant majority of users using registered accounts with respect to non-registered users.

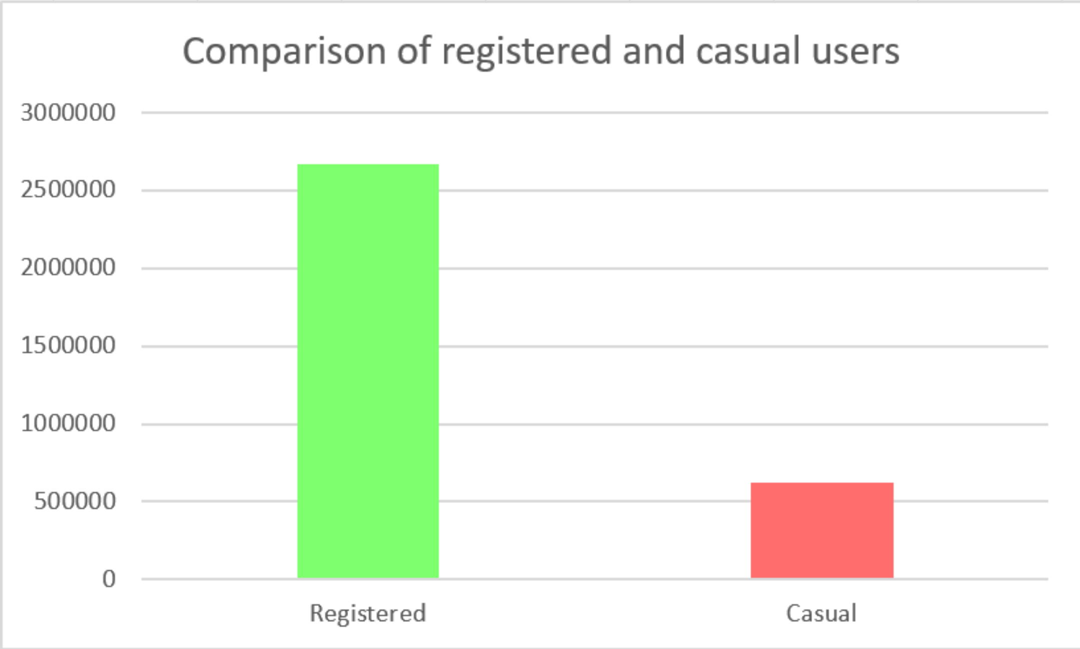
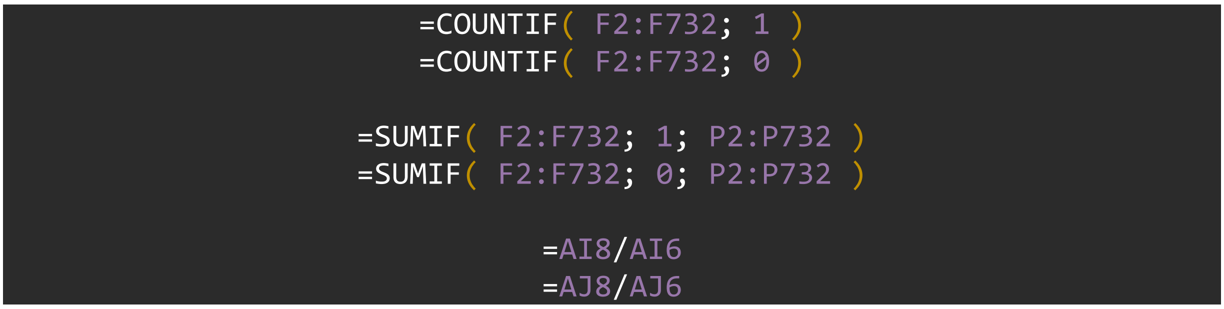


Figure 24 Comparison of registered and casual users analysis chart

The graph illustrates a huge discrepancy in the number of bikes rented by the group of registered and unregistered users. This discrepancy, as I mentioned earlier, does not directly have a major impact on the total number of rented city bikes, but in this case such results may be influenced by the situation where companies offering city bike rental services reward their users for creating an account and identifying themselves with it at the time of rental in the form of discounts on rental fees or increased time of city bike use without charging additional fees. Moreover, there may have been a situation where registered people have the ability to travel further distances on a rented city bike than unregistered people, who may have a limited perimeter of ability to travel on a rented city bike.

### 4.1.8 Impact of holidays on rentals analysis

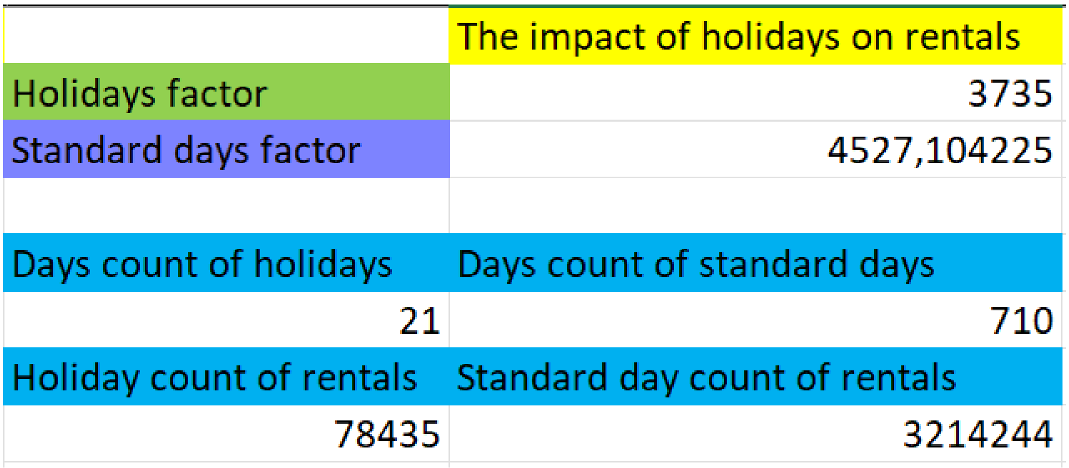
Analysis of the number of rented city bikes based on holidays and non-holidays is a very interesting statement. With the help of such an analysis, we are able to draw conclusions as to whether holidays and non-holiday days have any impact on the number of rented city bikes.



```
=COUNTIF( F2:F732; 1 )
=COUNTIF( F2:F732; 0 )

=SUMIF( F2:F732; 1; P2:P732 )
=SUMIF( F2:F732; 0; P2:P732 )

=AI8/AI6
=AJ8/AJ6
```

Figure 25 Impact of holidays on rentals analysis formulas

To carry out this analysis, I used three formulas, "COUNTIF", "SUMIF" and a formula that is a standard mathematical operation, in this case it is division. Using the "COUNTIF" formula, I calculated the number of occurrences of each type of day. The first segment of this formula is a range of cells containing information about whether a particular day is a holiday and is then marked with the number 1 or non-holiday, in which case the value of this entry for the "holiday" attribute is 0. The second segment, on the other hand, is to categorize and put a condition that must be met in order for the "COUNTIF" formula to know which values should be counted. Then, using the "SUMIF" formula, I calculated the exact number of city bikes rented for each type of day among holidays and non-holidays. Finally, the final result is the average number of rented city bikes for one holiday and non-holiday day. I obtained this value by dividing the sum of rented city bikes for a specific group among holiday and non-holiday days by the number of days on which each group occurs.



| | The impact of holidays on rentals |
|---|---|
| Holidays factor | 3735 |
| Standard days factor | 4527,104225 |
| | |
| Days count of holidays | Days count of standard days |
| 21 | 710 |
| Holiday count of rentals | Standard day count of rentals |
| 78435 | 3214244 |

**Figure 26 Output values of impact of holidays on rentals analysis**

This is a perfect example of when averaging values gives much more accurate information about the problem under study. In this case, we can see a huge discrepancy in the values representing the sum of all urban bicycle rentals for each group. In contrast, after calculating the average value of rented bicycles for a single day among each of the groups of holiday and non-holiday days analyzed, the difference is actually small.

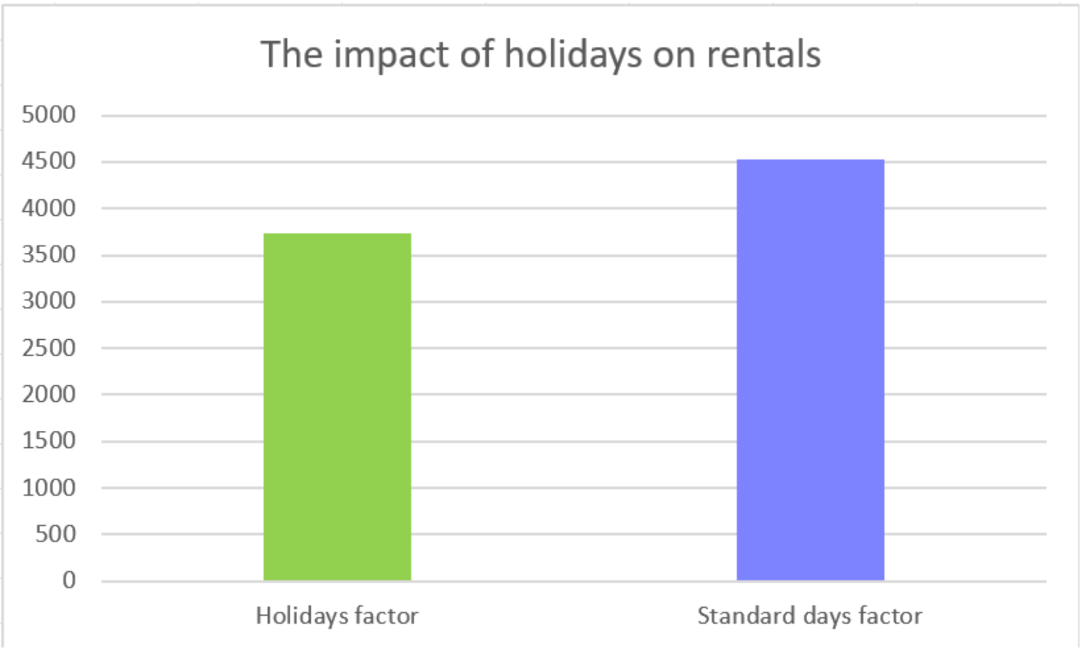


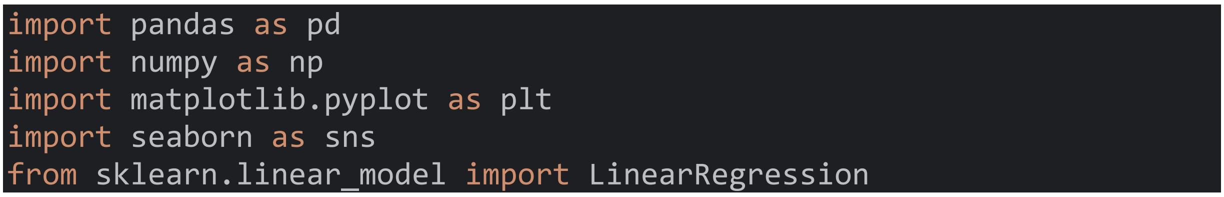
Figure 27 Impact of holidays on rentals analysis chart

The graphical projection prepared in the form of a chart illustrates very well how little difference there is in the average rental of city bicycles for a single day among holiday and non-holiday days. However, the small but still existing difference in these values could be the result of spending time on other duties related to the celebrated holiday. For example, it's easier to imagine a user renting a city bike to get to the store or any other destination than a user renting a city bike to get to the Christmas Eve celebration. On the other hand, based on the resulting data obtained from the analysis, such cases may occur.

## 4.2 Implementation of automated analysis

The program, which generates the resulting data and various types of graphs, was written in the "Python" programming language. In addition to the basic functions available in the "raw" programming language "Python", the program also uses additionally implemented five external libraries.

### 4.2.1 Library import implementation

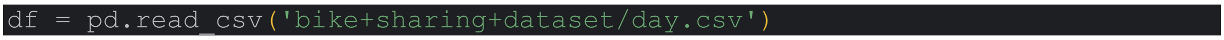


```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
```

Code 1 Library import

The "pandas" library in this program will be responsible for loading data, processing data, cleaning data and analyzing data. Moreover, using this library, the collected data will be prepared directly for training the machine learning model. The "numpy" library has several key roles for the operation of the program. It is mainly responsible for numerical calculations and data processing. In addition, it provides seamless integration with other libraries used in the program, such as "scikit-learn", "pandas", "matplotlib" and "seaborn". Next in order of imported libraries is the "matplotlib" library, which in this program is used to create visualizations of the resulting data. The advantage of using this library is that the graphs created by this library, allow you to easily understand how the different variables are related to each other. What's more, it can easily handle visualizations of complex dependencies in data, and the end result in the form of visualization of these dependencies is very friendly and easy to understand for the user reading the generated charts. The next library is the "seaborn" library, which is also used for data visualization. For example, with its use graphs depicting heat maps of the analyzed data were easily generated. The last library imported is the "scikit-learn" library, the use of which is directed at performing machine learning operations. In particular, an important element for this program is to enable the use of linear regression, which is, in fact, the basis of the automated analysis carried out by this program written in the "Python" programming language.

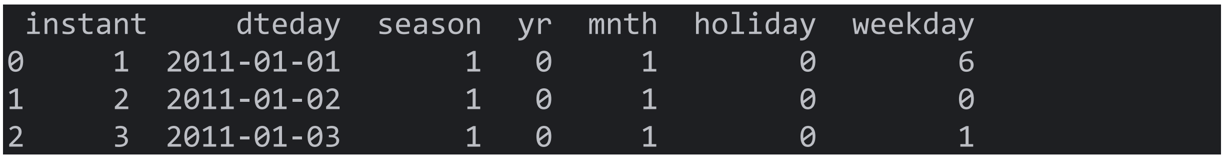
### 4.2.2 Data upload implementation



```
df = pd.read_csv('bike+sharing+dataset/day.csv')
```

Code 2 Data upload

The code for transferring data for use in the program is the first step to perform the analysis using a program written in the "Python" programming language. This line of code loads data from a file with the extension "CSV" into a "DataFrame" object using tools in the "pandas" library. The abbreviation "pd" is the abbreviation used for the "pandas" library, the declaration that allows the use of such an abbreviation was written when importing the "pandas" library. The ".read_csv" function is one of the offered tools included in the tools available in the "pandas" library, it is used to load data from files having a "CSV" extension. The parameter passed to this function is the path of the location where the "CSV" file to be loaded is stored. The ".read_csv" function returns a "DataFrame" object, which is stored as "df". The "Dataframe" object is a very flexible data structure that allows us to analyze, process and visualize data more easily. If we wanted to display the first few items of the prepared "DataFrame" object using the ".read_csv" function, it would look as shown below.



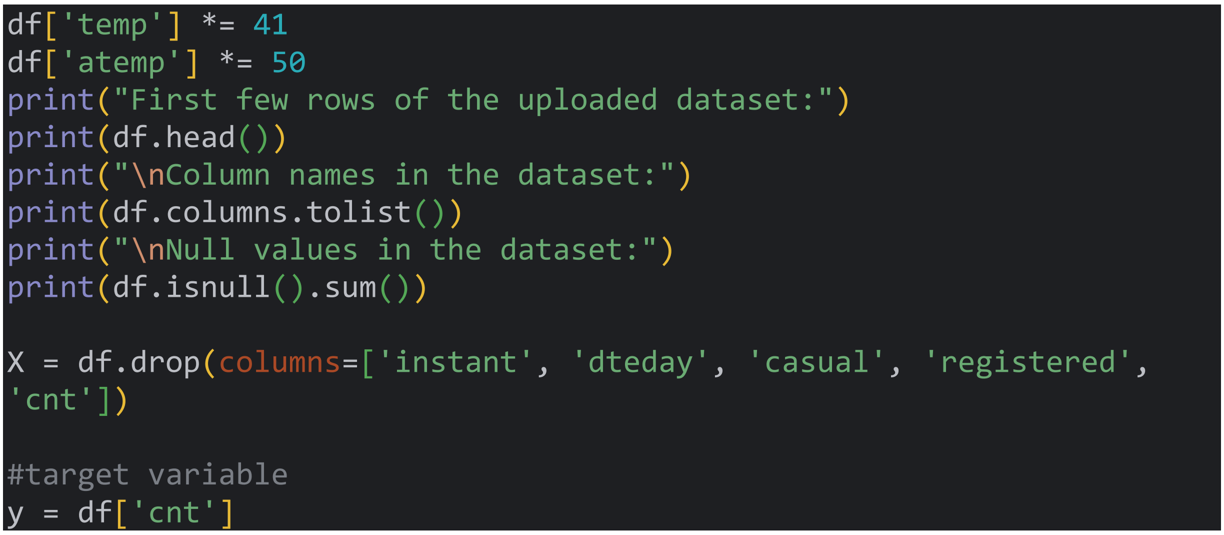
```
 instant     dteday  season  yr  mnth  holiday  weekday
0     1  2011-01-01       1   0     1        0        6
1     2  2011-01-02       1   0     1        0        0
2     3  2011-01-03       1   0     1        0        1
```

Figure 28 DataFrame structure example

The "DataFrame" object containing the data read from the "CSV" file will allow further analysis.

### 4.2.3 Implementation of data preparation

Proper preparation of data before further analysis is crucial in machine learning.



```
df['temp'] *= 41
df['atemp'] *= 50
print("First few rows of the uploaded dataset:")
print(df.head())
print("\nColumn names in the dataset:")
print(df.columns.tolist())
print("\nNull values in the dataset:")
print(df.isnull().sum())

X = df.drop(columns=['instant', 'dteday', 'casual', 'registered',
'cnt'])

#target variable
y = df['cnt']
```

Code 3 Data preparation

In the code snippet shown above, the preparation of the data contained in the "DataFrame" object was carried out. The column described by the "temp" attribute contains normalized temperature values that are scaled by the maximum value of 41, according to the data description attached to the dataset. Each value in the column described by the "temp" attribute is multiplied by the number 41 to convert it to the actual temperature in degrees Celsius. A similar modification has been applied to the column described by the "atemp" attribute, representing the perceived temperature, while the values assigned to this column have been multiplied by the value 50, adequate to the information provided in the dataset description. Next, columns that are not needed to train the model are removed. These include columns described by the attributes "instant", "dteday", "casual", "registered" and "cnt". The "instant" column is an index of an entry in the dataset, which is a unique identifier and carries no useful information for the machine learning model. The "dteday" column is a date in text format that is not directly used as a feature of the machine learning model, but based on this column, there are other columns in the dataset whose use affects the analysis. Such a column, for example, is the one marked with the "yr" attribute, which contains information about the year among the two-year period in which a particular entry in the dataset was recorded. Columns marked with the "casual" and "registered" attributes were also discarded from the target dataset submitted to the machine learning model, as both columns represent the number of recorded bicycle rentals for each of the registered and unregistered user groups. The sum of these values represents the target value denoted by the "cnt" attribute, which was also discarded from the transmitted dataset presented as a "DataFrame" object under the "X" variable. On the other hand, the "cnt" column was marked as the target value for the machine learning model and stored under the "Y" variable.

### 4.2.4 Modeling implementation

By training the linear regression model on the entire dataset, accurate results can be obtained that show how different characteristics affect the target variable "cnt."

```
#train the linear regression model using the whole dataset
model = LinearRegression()
model.fit(X, y)
```

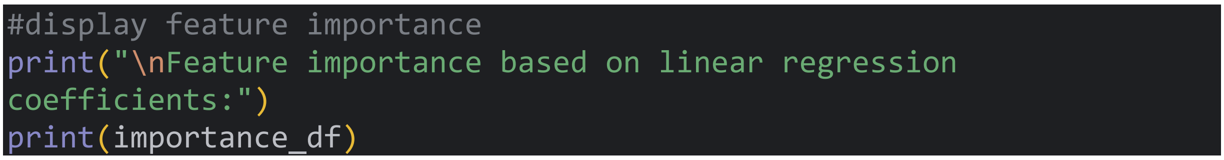
Code 4 Modeling

The "scikit-learn" library contains an implementation of various machine learning algorithms, including linear regression. The function "LinearRegression" is a class that implements linear regression. Using this function and assigning it to the variable "model", creates an instance of a linear regression model. The model is then trained on the input data "X", which is a selected and reprocessed data set in the form of a "DataFrame" object, and the output data, which is the target data assigned under the variable "Y". The model learns how different characteristics affect the number of rented bicycles.

### 4.2.5 Implementation of model results

Segments of the model training and results display code create a linear regression model, train it on the bike rental data, and then calculate and display the importance of each feature. This makes it possible to understand which characteristics have the greatest impact on the number of bicycle rentals, which can be useful, for example, when making decisions about marketing or operational strategies.

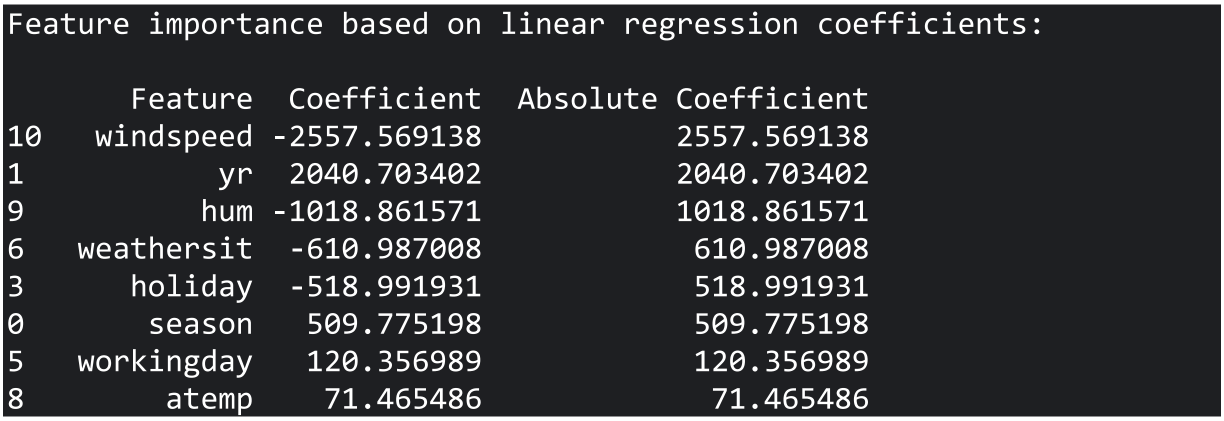
```
#get the coefficients and their corresponding headers
coefficients = model.coef_
features = X.columns

#create a df to display feature importance
importance_df = pd.DataFrame({'Feature': features, 'Coefficient':
coefficients})
importance_df['Absolute Coefficient'] =
np.abs(importance_df['Coefficient'])
importance_df = importance_df.sort_values(by='Absolute Coefficient',
ascending=False)
```



```
#display feature importance
print("\nFeature importance based on linear regression
coefficients:")
print(importance_df)
```

Code 5 Model results

The variable "coefficients" from "model.coef_" returns an array of regression coefficients for each feature. These coefficients indicate how each feature affects the target value "cnt". Another "DataFrame" object is then created, containing the feature names assigned under the "features" variable and the corresponding linear regression coefficients stored under the "coefficients" variable. The regression coefficients provide valuable information about the influence of individual features on the number of rented bicycles. A higher coefficient indicates a greater influence of a given characteristic. While coefficients can be a negative value, this indicates a negative impact on the number of rented bicycles. To further illustrate how much a particular characteristic affects the target value representing the number of rented bicycles, an absolute value releasing function was used. This function belongs to the tools provided by the "numpy" library. The prepared "DataFrame" object containing the coefficients of linear regression and the corresponding absolute values is printed out using the "print()" function, which is a tool of the "Python" language.

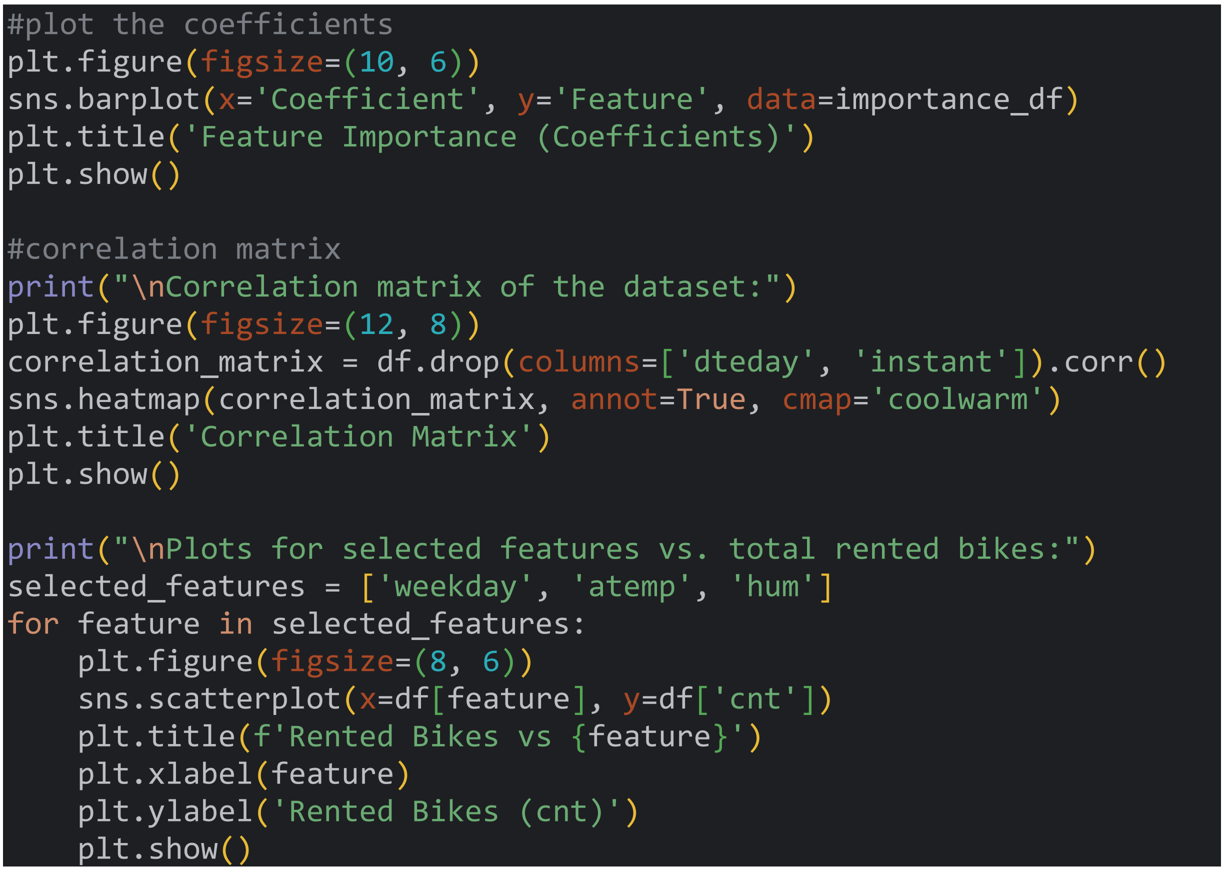


```
Feature importance based on linear regression coefficients:

       Feature  Coefficient  Absolute Coefficient
10   windspeed -2557.569138           2557.569138
1           yr  2040.703402           2040.703402
9          hum -1018.861571           1018.861571
6   weathersit  -610.987008            610.987008
3      holiday  -518.991931            518.991931
0       season   509.775198            509.775198
5   workingday   120.356989            120.356989
8        atemp    71.465486             71.465486
```

Figure 29 Feature importance based on linear regression coefficients

### 4.2.6 Implementation of visualization of results

The last segment of code included in the program written in the "Python" programming language for visualizing the results creates graphs that help you understand the meaning of the different features and the connections between them. With bar, heat and dot plots, it is possible to quickly and intuitively identify the most important features and understand how different features affect each other and the number of urban bicycles rented.



```
#plot the coefficients
plt.figure(figsize=(10, 6))
sns.barplot(x='Coefficient', y='Feature', data=importance_df)
plt.title('Feature Importance (Coefficients)')
plt.show()

#correlation matrix
print("\nCorrelation matrix of the dataset:")
plt.figure(figsize=(12, 8))
correlation_matrix = df.drop(columns=['dteday', 'instant']).corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()

print("\nPlots for selected features vs. total rented bikes:")
selected_features = ['weekday', 'atemp', 'hum']
for feature in selected_features:
    plt.figure(figsize=(8, 6))
    sns.scatterplot(x=df[feature], y=df['cnt'])
    plt.title(f'Rented Bikes vs {feature}')
    plt.xlabel(feature)
    plt.ylabel('Rented Bikes (cnt)')
    plt.show()
```

Code 6 Visualization of results

The "figure()" function, which is a tool from the "matplotlib" library toolbox, controls the parameters of the graph. In the case shown above, the size of the graph is declared. With the help of tools available in the "seaborn" library, such as "barplot()" it is possible to create bar charts, in this case the values of regression coefficients are shown on the abscissa axis, and the ordinate axis contains the names of features. The "show()" function displays the graph prepared to the user's requirements.

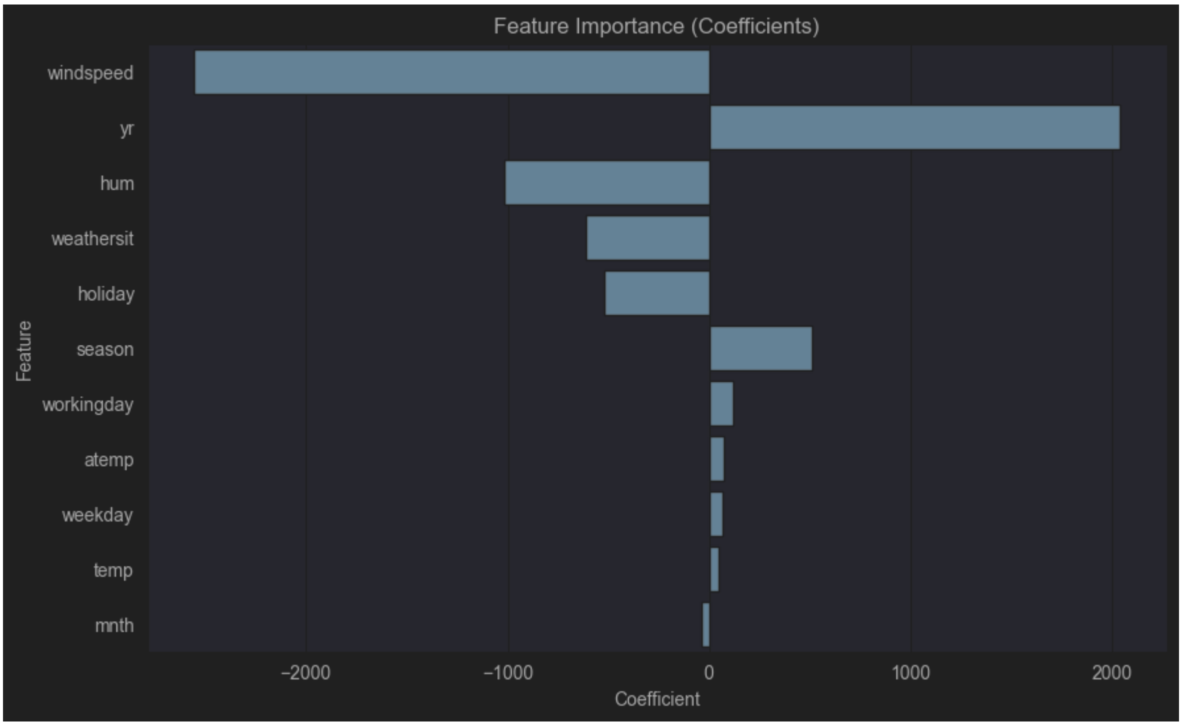


**Figure 30 Feature importance chart**

Analyzing the chart above, we are able to see that the value in the dataset that has the greatest negative impact on the number of urban bike rentals over the two-year period recorded in the "Bike Sharing" dataset is the column described by the "windspeed" attribute, which contains data on wind speed. In contrast, the most favorable values are the data described by the "yr" attribute. Using this graph, it can be concluded that the wind speed increases, the number of rented bicycles decreases, and the factor most favorable to the number of rented bicycles is time. The next graphical visualization generated is the correlation matrix, which is a heat map. A heatmap of the correlation matrix shows how individual features are related to each other. This can help identify characteristics that are highly correlated with the number of rented bicycles and with each other.

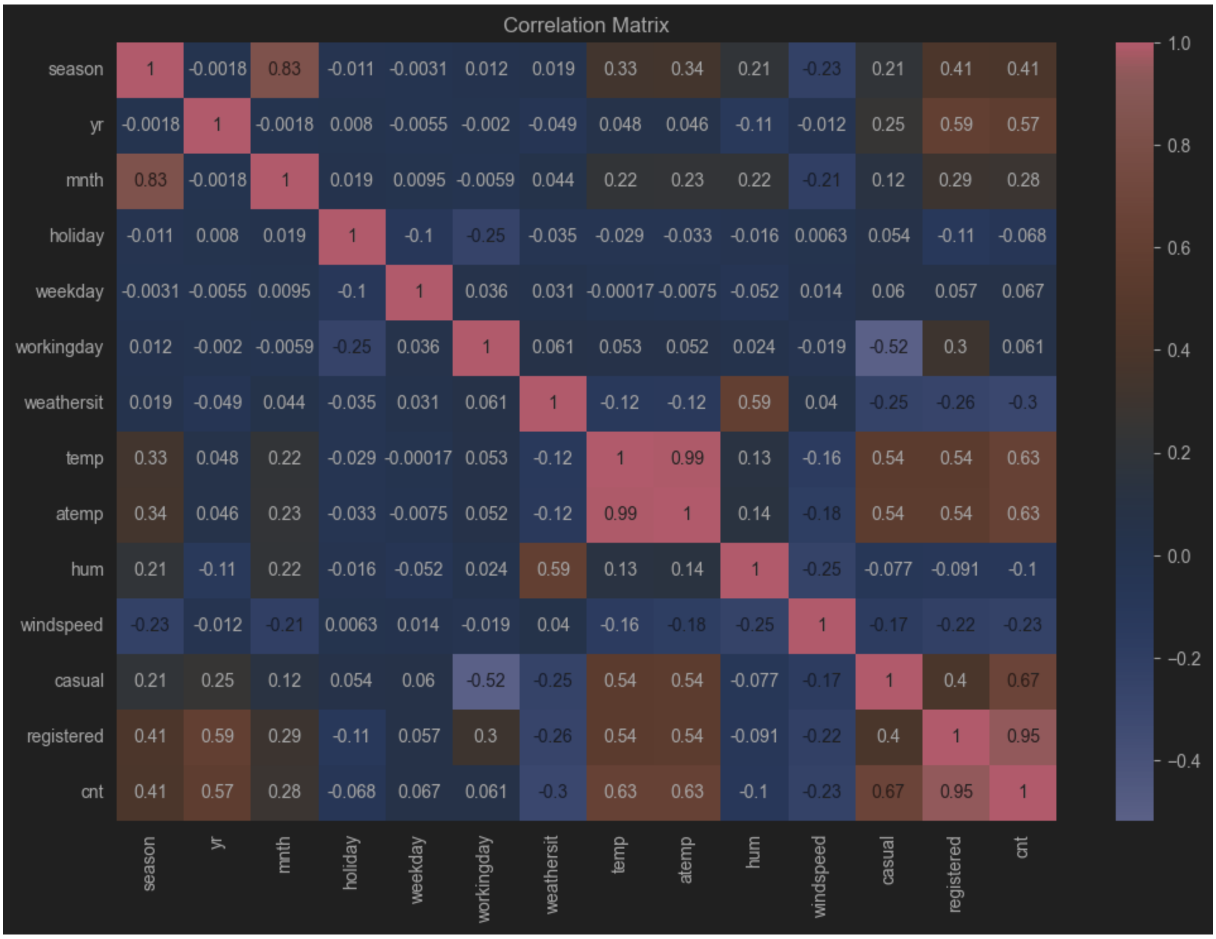


**Figure 31 Chart of correlation matrix**

Analyzing the chart above, we can come to some interesting conclusions. The important thing to note from this graph is that both the "temp" and "atemp" characteristics have a key impact on the number of urban bicycles rented. This result suggests that people are more likely to rent bikes on warmer days. Season and year also have a significant impact on the number of urban bicycle rentals, which may be related to seasonal activity patterns and trends in the growth of bicycle rental popularity. Weather conditions and wind speed have a negative impact on the number of urban bicycle rentals, indicating that worse conditions discourage potential users from using this mode of transportation. Interestingly, the "casual" coefficient also has a high positive correlation with "cnt" of 0.67, but not as strong as the "registered" coefficient, whose correlation with "cnt" is as high as 0.95. This means that the number of registered users is strongly related to the total number of rented bicycles.

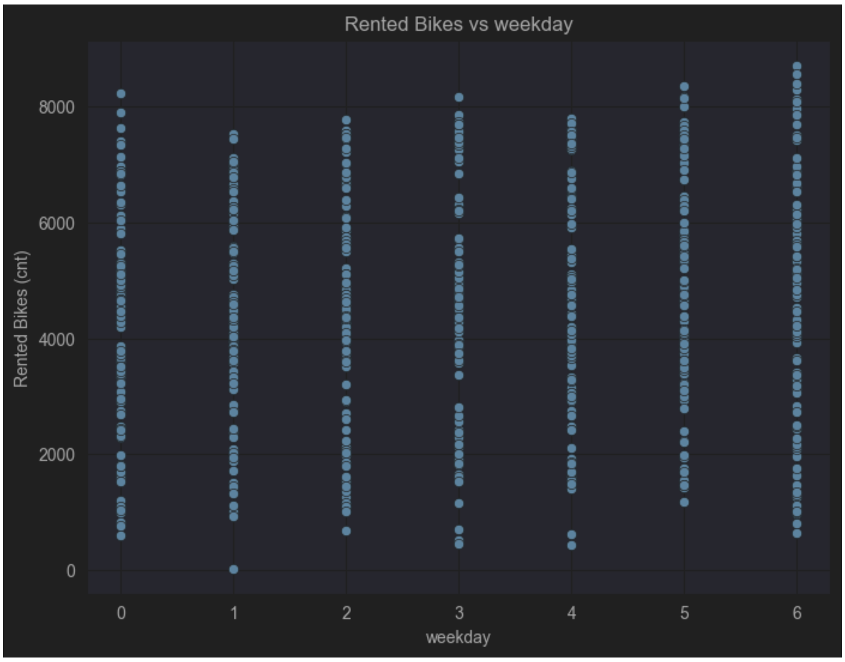


Figure 32 Weekday and rented bikes chart

This chart shows the number of bicycle rentals by day of the week. It can be seen that rentals are relatively evenly distributed throughout the week. There are no clear differences in the number of urban bicycles rented between different days of the week, suggesting that the day of the week does not have much impact on the number of bikes rented.

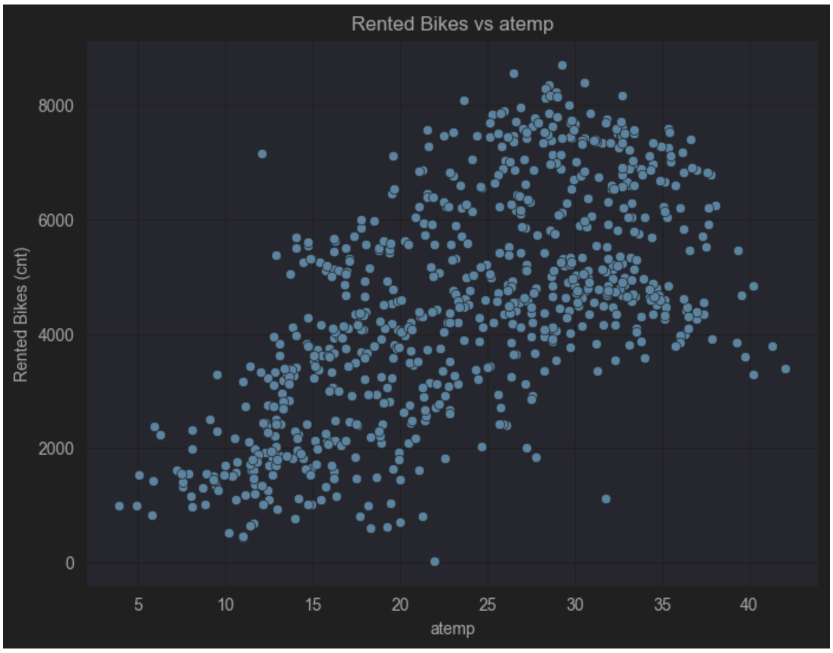


Figure 33 Feeling temperature and rented bikes chart

The graph presented here shows the dependence of the number of rented bicycles on the perceptible temperature. There is a clear positive correlation showing that the higher the perceptible temperature, the higher the number of rented bicycles. This suggests that warmer weather is conducive to a higher number of bicycle rentals.

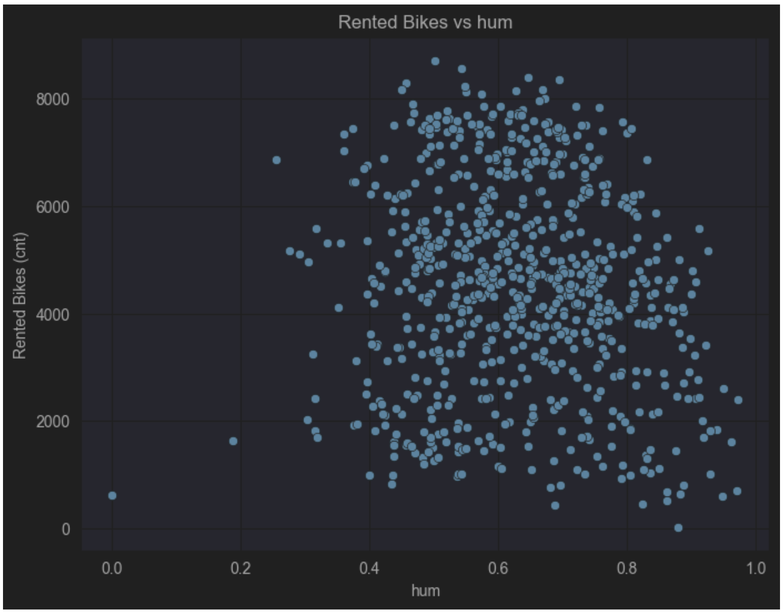


Figure 34 Humidity and rented bikes chart

The graph shows the relationship between the number of bicycles rented and humidity. There is no clear correlation between these variables. Rentals appear to be fairly evenly distributed regardless of humidity levels, suggesting that humidity does not have a significant effect on the number of bicycles rented.

# 5 Summary

## 5.1 Analysis outcome

The "Bike Sharing" dataset analysis project aimed to understand what factors affect the number of rentals in a given urban bike rental system. Various variables such as weather, day of the week, holidays, season, month, humidity and wind speed were analyzed to determine their impact on rentals. The final element of the analysis performed is the data visualizations, which are a key element of the analysis as they allow for easy understanding of complex relationships. Analysis of the data from the "Bike Sharing" dataset showed that the most important factor affecting the number of bicycle rentals is the weather, which includes several factors such as temperature, both actual and perceived, and wind speed. Day of the week and humidity are less important. This information can be useful to operators of bicycle rental systems, helping them to better understand what conditions are conducive to higher bicycle use, and to adjust operational activities according to these conditions.

## 5.2 Acknowledgments

The most interesting and, at the same time, the most rewarding part of conducting this analysis with its documentation was the expansion of my analytical knowledge. It helped me to look at many things happening around me from a different angle. A huge thank you goes to postdoctoral fellow Grzegorz Marcin Wojcik, who showed incredible professionalism and experience in data analysis as well as in his role as a supervisor. I also thank him for providing a lot of valuable advice and guidance during the writing of the thesis. The knowledge and experience he imparted is invaluable, and his help in managing this project resulted in the proper development of the thesis. Also, I would like to thank all the people close to me, my parents, my friends for providing me with a place to work, for supporting me in hard times, for valuable advice in moments of brainstorming.

# 6 Bibliography

- *Alan Turing*. (2024, May 30). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Alan_Turing
- *Analysis*. (2024, May 28). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Analysis
- *Arthur Samuel*. (2024, June 12). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Arthur_Samuel_(computer_scientist)
- *Bicycle-sharing system*. (2024, April 19). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Bicycle-sharing_system
- *Code refactoring*. (2024, June 8). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Code_refactoring
- *Data analysis*. (2024, May 20). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Data_analysis#:~:text=Data%20analysis%20is%20a%20process,test%20hypotheses%2C%20or%20disprove%20theories.
- Flower, M., Beck, K., Opdyke, W., & Brant, J. (2002). Refactoring Helps You Find Bugs. In M. Flower, *Refactoring: Improving the design of Existing Code* (pp. 48-49). Addison-Wesley.
- Foote, K. D. (2021, December 3). *A Brief History of Machine Learning*. Retrieved from Dataversity: https://www.dataversity.net/a-brief-history-of-machine-learning/
- *Frank Rosenblatt*. (2024, May 6). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Frank_Rosenblatt
- Jetbrains. (2024). *IntelliJ IDEA overview*. Retrieved from https://www.jetbrains.com/help/idea/discover-intellij-idea.html
- LIAAD. (2024). *LIAAD*. Retrieved from LIAAD: http://www.liaad.up.pt/
- *Matplotlib*. (2024, May 9). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Matplotlib
- Microsoft. (2024). *Excel*. Retrieved from Excel — pomoc i informacje: https://support.microsoft.com/pl-pl/excel
- *NumPy*. (2024, April 2). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/NumPy
- Oxford, U. (2019, October 21). What is Machine Learning? *American Journal of Epidemiology*.

- Pandas. (2024). *Pandas*. Retrieved from https://pandas.pydata.org/
- Python Software Foundation. (2024). *Python*. Retrieved from https://www.python.org/about/
- Samuel, A. (1959). Some Studies in Machine Learning Using the Game of Checkers. In *IBM Journal of Research and Development* (pp. 210 - 229). IBM.
- *Scikit-learn*. (2024, May). Retrieved from scikit-learn: Machine Learning in Python: https://scikit-learn.org/stable/
- *Seaborn*. (2024). Retrieved from seaborn: statistical data visualization: https://seaborn.pydata.org/

# 7 Attachment

## 7.1 Table of Figures

## 7.2 Table of Pictures

## 7.3 Table of Code

## 7.4 What CD contains

- **\Szymon_Wujec_s20431_MasterThesis**
  - Electronic version of master’s thesis
- **\Data**
  - All necessary files used to formulate the master's thesis