Bike-sharing demand prediction

Ajit kumar patel

Data science trainee

Almabatter, Banglore

Featured application:

The main application of this work is the analysis and prediction of the demand in bike-sharing systems using their open/private data. A multi-agent system is proposed and a case study is conducted using the data of a bicycle sharing system from a middle size city.

abstract: This paper provides visualization and prediction tools for bike-sharing systems (BSS). The presented multi-agent system includes an agent that performs data collection and cleaning processes, it is also capable of creating demand forecasting models for each bicycle station. Moreover, the architecture offers API (Application Programming Interface) services and provides a web application for visualization and forecasting. This work aims to make the system generic enough for it to be able to integrate data from different types of bike-sharing systems. Thus, in future studies, it will be possible to employ the proposed system in different types of bike-sharing systems. This article contains a literature review, a section on the process of developing the system, and the built-in prediction models. Moreover, a case study validates the proposed system by implementing it in a public bicycle sharing system in Salamanca called SalenBici. It also includes an outline of the results and conclusions, a discussion on the challenges encountered in this domain, as well as possibilities for future work.

1.introduction::

There is a consensus in the literature that states that bicycles are one of the most sustainable modes of urban transport and they are suitable for both short trips and medium-distance trips. Riding a bicycle does not have any negative impact on the environment, it promotes physical activity and improves health. Furthermore, its use is cost-effective from the perspective of users and infrastructure. Moreover, due to the increased CO2 levels, the European Union and other states are taking measures to reduce greenhouse gas emissions in every sector of the economy. These facts explain the growing popularity of sustainable means of transport such as bike-sharing systems. From 1965 when they came into use in Amsterdam to 2001, there were only a few systems around the world. Bike-sharing systems (BSS) began to spread in 2012 when their number increased to over 400. By 2014 this number had doubled and nowadays there are approximately 1175 cities, municipalities, or district jurisdictions in 63 different countries where these systems are in active use, according to BikeSharingMap.

Bike-sharing systems allow users to travel in the city at a low cost or even for free. They can pick up a bicycle at one of the stations distributed across the city and leave it at another. These systems have evolved [8] and today the vast majority include sensors that provide information on the interaction of users with the system. However, the management of these systems and the data collected by them is often poor and as a result, the numbers of bicycles available at stations are not sufficient. These are the reasons why bike-sharing systems should be improved with data produced by the systems themselves. They should include predictive models for user behavior and demand, which will notify the system administrator of the stations where more bicycles are required for satisfying user demand. This will also allow to set up new stations in places where the demand is high or, on the contrary, to close down the stations at which the demand is too low.

2. BIKE SHARING SYSTEM:

All bike-sharing systems operate based on a common philosophy, their principle is simple: individuals use bicycles on an “as-needed” basis without the costs and responsibilities that owning a bicycle normally entail. a classical bike station, where there are free docks and available bikes to rent within a station map of Madrid’s Bike Sharing System called BiciMad. As shown on the map, the bike-sharing system offers a wide list of stations located across the center of the city. Peter Midgley indicates in his study that bike-sharing systems can be categorized into 4 generations depending on their features: (1) First-generation bike-sharing systems: the first generation of bike-sharing systems was introduced in Amsterdam (1965), La Rochelle (1976), and Cambridge (1993). These systems provided free bicycles which could be picked up and returned to any location. The vast majority of these systems were closed due to vandalism. (2) Second generation: this generation tries to solve the drawbacks of the previous one. In this case, the systems had a coin deposit (like the supermarket trolleys) but they still suffered from thefts due to the anonymity of the users. (3) Third generation: this generation uses high-tech solutions including electronic locking docks, smart cards, mobile applications, built-in GPS devices in the bikes, and totem applications. (4) Fourth generation: it is still in the process of development, this generation includes movable docking stations, solar-powered docking stations, electric bikes, and real-time system data.

6. problem descreption:

Currently Rental bikes are introduced in many urban cities for the enhancement of mobility comfort. It is important to make the rental bike available and accessible to the public at the right time as it lessens the waiting time. Eventually, providing the city with a stable supply of rental bikes becomes a major concern. The crucial part is the prediction of bike count required at each hour for the stable supply of rental bikes.

The dataset contains weather information (Temperature, Humidity, Windspeed, Visibility, Dewpoint, Solar radiation, Snowfall, Rainfall), the number of bikes rented per hour and date information.

**Attribute Information:**

Date : year-month-day

Rented Bike count - Count of bikes rented at each hour

Hour - Hour of he day

Temperature-Temperature in Celsius

Humidity - %

Windspeed - m/s

Visibility - 10m

Dew point temperature - Celsius

Solar radiation - MJ/m2

Rainfall - mm

Snowfall - cm

Seasons - Winter, Spring, Summer, Autumn

Holiday - Holiday/No holiday

Functional Day - NoFunc(Non Functional Hours), Fun(Functional hours)

2. BIKE-SHARING MAIN ISSUES:

Current literature lists factors that influence the success of bike-sharing systems, these include the built environment, psychological factors related to the natural environment, as well as the utility theory. This last reason focuses on providing users with the best quality of service, by addressing any issues encountered in the bike-sharing systems. Vogel et al. [9] describe three main issues in bike-sharing systems; their design, management, and operation. The authors distinguish three categories:

* Network design and redesign issues: These decisions address issues related to the initial design of the system, they consider the topography, traffic, and equity of stations located across the city [10]. The number of docks and the number of vehicles at each station is important factors in this category. These issues occur not only in the initial design but also in the subsequent use of the system:
* Incentivizing users to balance the system: These decisions are related to operational matters and they aim to mitigate the main operational problem: the shortage event [13] that occurs when a customer wants to pick up a bike from an empty station or return it to a station whose docks are occupied. Users are incentivized to occupy the stations with available bikes and docks.
* Operational reallocation issues: This category also focuses on an operational aspect, known as the commutation pattern in bike-sharing systems. There are specific bike station usage patterns, for example, in the mornings the stations located on the outskirts of cities are empty because many people travel to the city to work. On the other hand, the stations in the city tend to be full in the mornings [14]. As we mentioned previously, the empty and full stations must be balanced by system administrators and this reallocation must follow a specific strategy that is based on the demand predicted for each station provided by prediction demand models. They must additionally use a reallocation algorithm that minimizes the total cost of managing bike fleets. In this case, there are two more problems: the prediction engine and the optimization algorithm used to reallocate the bikes. This work addresses the prediction problem but does not discuss the reallocation strategy

3. PROPOSED SYSTEM:

This section details the proposed multi-agent system (MAS) and presents a general diagram of the system’s architecture which is shown in Figure 2. Literature shows that multi-agent systems have previously been employed in similar tasks like taxi fleet coordination where they offer an ideal solution to abstract away issues of the different existing platforms and communication protocols. The system is divided into the following groups of agents: bike-sharing data agents, weather data collector agents, geographical information agents, data persistence agents, demand prediction agents, and API agents. Several multi-agent platforms such as SPADE, JADE, PANGEA, AROMAS, and brain were evaluated and the brain was finally selected for this system because of its ease of use. Furthermore, it is implemented in Python (like SPADE and AROMAS) and it is in continuous development nowadays

4. BIKE DATA SHARING AGENTS:

Bike Data Sharing Agents are responsible for obtaining data from the BSS, which usually offers two kinds of data:

* Station information: It indicates the total number of available docks and bicycles at each station at a given time, that is, the current situation at the station. This data is usually available and provided to the end-users to keep them informed about the stations with available bicycles. If this data is periodically polled, pickups and returns at each station can be calculated regularly.
* • Trip information: This sort of information is not commonly published but it is usually available in CSV files offered every month as open data. It is related to the trips recorded by the BSS: a user picks a bike at a specific station at a given time and later returns it to another station. This kind of data discloses more details as it provides insights into the flow of bikes between stations and the users who perform the trips recorded in the system

5. DATA & collection process:

The data collected regards all trips traveled in the system from a station of origin to a final station. In this case, the information provided by SalenBici company is trip data, as mentioned previously and it has the following information in the original dataset: (1) Time Start: timestamp of the beginning of the trip, (2) Time End: timestamp of the end of the trip, (3) Bicycle ID: unique bike identifier, (4) Origin Station: origin station name, (5) End Station: destination station name, (6) Origin dock: origin dock identifier, (7) End dock: destination dock identifier, (8) User ID: user unique identifiers,

6. steps involved:

* **Exploratory Data Analysis**

After loading the dataset we performed this method by comparing our target variable that is rent\_count with other independent variables. This process helped us figuring out various aspects and relationships among the target and the independent variables. It gave us a better idea of which feature behaves in which manner compared to the target variable.

* **Null values Treatment**

Our dataset does not contains the null nither which might tend to disturb our accuracy hence we dropped them at the beginning of our project inorder to get a better result.

* **Encoding of categorical columns**

We used One Hot Encoding to produce binary integers of 0 and 1 to encode our categorical features because categorical features that are in string format cannot be understood by the machine and needs to be converted to numerical format.

* **Feature Selection**

In these steps we used algorithms like ExtraTree classifier to check the results of each feature i.e which feature is more important compared to our model and which is of less importance.

Next we used Chi2 for categorical features and ANOVA for numerical features to select the best feature which we will be using further in our model.

* **Standardization of features**

Our main motive through this step was to scale our data into a uniform format that would allow us to utilize the data in a better way while performing fitting and applying different algorithms to it.

The basic goal was to enforce a level of consistency or uniformity to certain practices or operations within the selected environment.

7. MODEL SELACTION:

This section describes how the BSS dataset was split and the methodology that was used in order to select the prediction models that will be included in the predictor agent. Like in Kaggle competitions [17], the data have been split into two datasets; a training dataset and a validation dataset. Figure 8 shows schematically how the available data were employed in the training, selection and validation of the models used. In the upper part of the figure, the green part represents the entire dataset. Like in Kaggle competition, a validation dataset is initially extracted and it is formed from the 20th to the end of each month, in the diagram it is represented in blue. The rest of the dataset, (those from the 1st to the 19th of each month), will be used as training data, represented in violet in the diagram. These data will be one of the inputs of the hyperparameter search technique: GridSearchCV [50]. This technique will use the following as inputs: (1) regression algorithms with their corresponding parameter grid; (2) a scoring function, in order to evaluate the input models, in this case RMSLE and R2 ; finally; (3) a cross validation method, in this case TimeSeriesSplit, a method that is intended specifically for time series and which resembles the usual functioning of a system in production. This method makes it possible to progressively use data from the past for training and use future data for validation, a diagram showing its functioning g is situated in the lower part under GridSearchCV.

The GridSearch method will make all the possible combinations for each algorithm with the provided grid parameters; this will be done by employing the cross-validation method (Time Series Split) and evaluating the trained methods with the provided scoring functions. As the output of this method, models for each algorithm with the best results will be obtained and these will be evaluated with the validation dataset that had been split at the beginning, on the right-hand side of Figure 8. A Dummy Regressor has been added to the models used and was established as their prediction strategy in order to continually predict the average. The regression algorithms as well as the following parameter girds have been trained using GridSearchCV

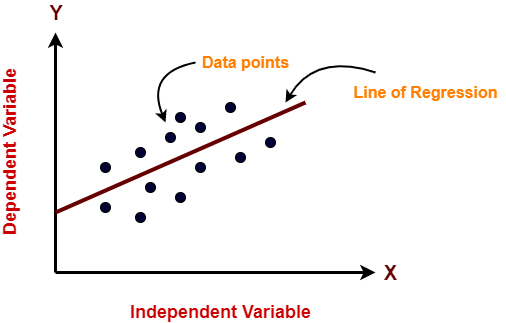
* **Fitting different models**

For modelling we tried various classification algorithms like:

1. **Linear Regression**
2. **Lasso regression**
3. **Ridge regression**
4. **Elasticsnet regression**
5. **Decession tree**
6. **Random Forest Classifier**
7. **Gradinet boosting**

7. ALGORITHM:

* **LINEAR REGRESSION:**
* Linear Regression is an ML algorithm used for supervised learning. Linear regression performs the task to predict a dependent variable(target) based on the given independent variable(s). So, this regression technique finds out a linear relationship between a dependent variable and the other given independent variables. Hence, the name of this algorithm is Linear Regression.

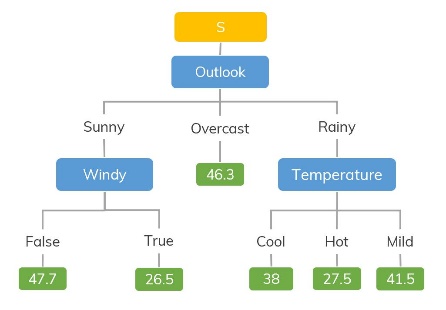
****

In the figure above, on X-axis is the independent variable and on Y-axis is the output. The regression line is the best fit line for a model. And our main objective in this algorithm is to find this best fit line.

**Pros:**

* Linear Regression is simple to implement.
* Less complexity compared to other algorithms.
* Linear Regression may lead to over-fitting but it can be avoided using some dimensionality reduction techniques, regularization techniques, and cross-validation.

**Cons:**

* Outliers affect this algorithm badly.
* It over-simplifies real-world problems by assuming a linear relationship among the variables, hence not recommended for practical use-cases.
* **DECISION TREE:**
* The decision tree models can be applied to all those data which contains numerical features and categorical features. Decision trees are good at capturing non-linear interaction between the features and the target variable. Decision trees somewhat match human-level thinking so it’s very intuitive to understand the data.
* ****

For example, if we are classifying how many hours a kid plays in particular weather then the decision tree looks like somewhat this above in the image.

So, in short, a decision tree is a tree where each node represents a feature, each branch represents a decision, and each leaf represents an outcome(numerical value for regression).

**Pros:**

* Easy to understand and interpret, visually intuitive.
* It can work with numerical and categorical features.
* Requires little data preprocessing: no need for one-hot encoding, dummy variables, etc.

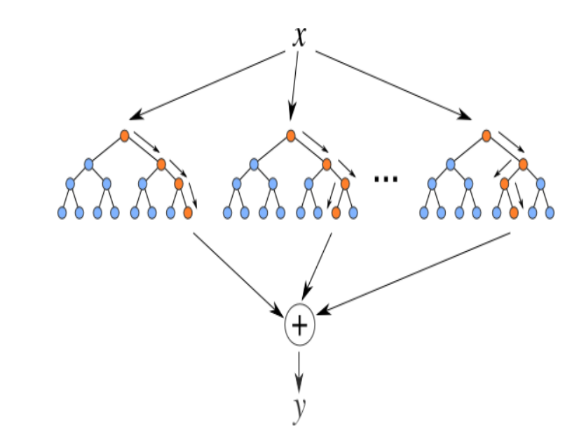
**Cons:**

* It tends to overfit.
* A small change in the data tends to cause a big difference in the tree structure, which causes instability.
* **LASSO REGRESSION:**
* LASSO stands for Least Absolute Selection Shrinkage Operator. Shrinkage is basically defined as a constraint on attributes or parameters.
* The algorithm operates by finding and applying a constraint on the model attributes that cause regression coefficients for some variables to shrink toward a zero.
* Variables with a regression coefficient of zero are excluded from the model.
* So, lasso regression analysis is basically a shrinkage and variable selection method and it helps to determine which of the predictors are most important.

**Pros:**

* It avoids overfitting

**Cons:**

* LASSO will select only one feature from a group of correlated features
* Selected features can be highly biased.
* **RANDOM FOREST REGRESSOR:**
* Random Forests are an ensemble(combination) of decision trees. It is a Supervised Learning algorithm used for classification and regression. The input data is passed through multiple decision trees. It executes by constructing a different number of decision trees at training time and outputting the class that is the mode of the classes (for classification) or mean prediction (for regression) of the individual trees.

**Pros:**

* Good at learning complex and non-linear relationships
* Very easy to interpret and understand

**Cons:**

* They are prone to overfitting
* Using larger random forest ensembles to achieve higher performance slows down their speed and then they also need more memory.

8. CONCLUSION:

That's it! We reached the end of our exercise.

Starting with loading the data so far we have done EDA , encoding of categorical columns, feature selection and then model building.

In all of these models our accuracy revolves in the range of 90 to 95%.

And there is no such improvement in accuracy score even after hyperparameter tuning.

So the accuracy of our best model is 73% which can be said to be good for this large dataset. This performance could be due to various reasons like: no proper pattern of data, too much data, not enough relevant features.

* The Rented Bike Count has been increased from 2017 to 2018.
* No overfitting is seen.
* Random forest & gradient boosting greedsearch cv gives the highest R2 score of 98% ,91% for train and 86% and 87% for Test set respectively
* Feature Importance value for Random Forest, gradient boosting greedsearch cv are defferent
* We can deploy this model

**References-**

1. Researchgate.net
2. Google
3. Analytics Vidhya