

SEOUL BIKE DEMAND

Data analysis



Équipe :
TRANG THOMAS
NASSAR IBRAHIM

Supervisor :
M. BERTIN LUC



Contents

1	Context	2
2	Project steps	3
2.1	Preprocessing	3
2.2	Visualisation	3
2.3	Modeling	4
2.4	Results	5
2.5	Discussion	6
2.6	Transform the model to an API Flask	6
3	Acknowledgment	7
4	Conclusion	7



1 Context

Currently rental bikes are introduced in many urban cities for the enhancement of mobility comfort. The purpose of this movement is to modernize cities and encourage people to head to a green world. For example, in Paris 2007, "velibs" were introduced, and in Amsterdam, there are more bikes than cars. The goal is to facilitate commuting in the city and reduce the amount of cars and the pollution. Indeed, the development of the way to commute reduced the use of cars to go to work and visit the city.

It is important to make the rental bike available and accessible to the public, as it provides many alternatives to commuters in metropolises. There are a lot of advantages to bike rents, it is convenient because it permits people not to keep the bike all day long, whether it is at work or at school. Furthermore it is the healthiest way to travel and it has environmental benefits. The studied dataset contains weather information (Temperature, Humidity, Wind speed, Visibility, Dew point, Solar radiation, Snowfall, Rainfall), the target is the number of bikes rented per hour and date information. The dataset presents the company's data between December the 1st of 2017 and finishes one year later. This study could have many aims for the company that could be seeing the results of the past year. It could also help them ameliorate themselves to become better and have a full satisfaction from customers.

How could we know how many bikes are rented per hour in function of weather conditions ?

The goal of the company Seoul Bike is providing the city with a stable supply of rental bikes. It becomes a major concern to keep user satisfied. The crucial part is the prediction of bike count rents at each hour for a stable supply of rental bikes. We can suppose that this study could be reported to the company 'Seoul Bikes'. We think it could help them knowing if yes or not they have to supply bikes stations in the city, in order to keep a good satisfaction for the users.



2 Project steps

On the dataset, we have 1 year of data. It begins at 2017-12-01 and it ends one year later, 2018-11-30. The time series has a step of 1 hour, so we have 24 lines of data per day in our DataFrame. The target is the rented bike count per hour and the features are mainly weather features like temperature, wind, rain, snow but also the importance of the studied day.

2.1 Preprocessing

Data preprocessing is a data mining technique which consists in transforming the data in order to make it understandable. It could be changing the type, the format, splitting the data, verify that there are no missing values but also creating new columns thanks to columns we initially have. In machine learning, the data processing step is critical because it involves cleaning, integration, transformation, scaling, standardize data and many other tasks, in order to have a good preparation for the application of models.

To begin we first did some **data exploration** by checking types, missing values and data description. We also **changed the date type** to DateTime which was initially a str object. Then, we created a column which takes the hour of the day and returns if it the day of the week **night or day** moment of the day (because we remind you that data is collected by hour), in order to do data visualisation with the target. From the date, we also created two columns with the **day of the week** and the **month** of the year corresponding. And finally also did an encoding on the day to convert 'WeekDays' in order to visualise it and make it a feature for ML models.

2.2 Visualisation

The first plot is maybe one of the **most important** because it shows **all the correlations** of the features. To complete this visualisation, we created a ranking of the features which are the most correlated to the target. So it gives an idea of which features we have to focus on.

The following plot that we shows us **the sum of rented bikes month by month** in 2018. This distribution clearly shows us that there is a high raise of rents from April to November. That's why we decided to verify that the rents raised proportionally to the temperature and to the solar radiation. Moreover on the notebook, you can see that as expected **the raise of rainfall and snowfall** comes with a decrease of the rents which is totally logical.

We also **created** a feature : "Night/Day" in order to see the distribution following the moment of the day. **From 8pm to 5am**, we decided to qualify this moment as 'night' and the rest is 'day'.

Finally we wondered what were the hours during which the rents were the highest, and we found that it was around 8am and around 6pm which confirms that people take bikes to go to school or work and go home at night. This analysis is very interesting because it shows that **koreans** take the global warming **very seriously** and respect the earth.



Of course you can find all the visualisations on the notebook Jupyter. Let's continue with the modeling part which is going to show us how to set machine learning algorithm to predict then number of rented bikes from weather conditions.

2.3 Modeling

We have a regression problem because our target is the number of rented bikes per hour. So the goal of this part is to apply many algorithms in order to find the algorithm with the best indicator. The indicator we decided to choose is the R^2 . This choice is because we wanted to be able to compare these algorithms between them and to choose which one is the most efficient. Let's apply regression techniques to our problem.

- **Linear multiple regression**

We ran a multiple linear regression, we assume that all the features have a linear relationship with the target. We also have to assume that these features have a Gaussian distribution and that features are not highly correlated between them, it is called multi-co linearity. The goal of this model is to minimize the RMSE and to get the R^2 close to 1 or -1. We first fit the model on the training data and training target. Then we first predicted the training data then the test data in order to get indicators.

For the training set, we got a R^2 score = 0.48 and a RMSE = 468 which is not that huge knowing that we have 6394 values predicted. For the testing set, we got a R^2 score = 0.44 and a RMSE = 467 which is not a very good score.

The test and train score are very close.

- **Ridge and Lasso**

We first did the **Ridge regression**, and performed a grid search with different values of alpha. Then we calculated R^2 train and test as we have an indicator to compare it to the linear regression. For the training set, we got a highest R^2 score = 0.48. For the testing set, we got a R^2 score = 0.47.

These scores are very close to the previous model. That's why we are going to try other models. Then we tried the **Lasso Regression** which is a little bit different from Ridge regression. Unlike ridge, Lasso can exclude useless feature from the model because it reduces variance, so it helps do features selection. The Lasso regression can make features disappear because of the shape when we reduce dimension. The shape of the Lasso is a diamond whereas the Ridge is an ellipse. Then we performed a grid search on the Lasso for which we had the same of score as the Ridge grid search. We got 0.48 R^2 score.

- **Decision Tree**

Then we applied a decision tree regressor on our data. We first scaled our X training and X

testing sets. Then we applied a grid search on the model by tuning the feature 'max depth'. We fit our grid search on the X and y training sets. Then we kept the best model (with the best estimators) and got the score with the X and y testing sets. The result of the testing sets is 0.74 and is much more higher.

- **Random Forest regressor**

Then we applied a **random forest regressor** on our data. We also take the scaled X training and X testing sets. Then we made a **grid search** on the model by tuning the feature 'max depth', 'n estimators', 'min samples split', 'min samples leaf' and 'bootstrap'. We fit our grid search on the X and y training sets. Then we kept the best model (with **the best estimators**) and get the score with the X and y testing sets corresponding to the model with the best estimators. The result of the testing sets is 0.85 and is much higher and is a pretty good score.

- **Extra Trees Regressor**

Finally we applied an **extra trees regressor** on our data. For this model, we also take the scaled X training and X testing sets to fit the model. Then we made a **grid search** on the model by tuning the feature 'max depth', 'n estimators', 'min samples split', 'min samples leaf' and 'bootstrap'. We fit our grid search on the X and y training sets. Then we kept the best model (with the best estimators) and got the score with the X and y testing sets. The score is pretty similar but a little bit higher than the random forest regressor. The best result of the testing sets that we got is 0.86 which is a pretty good score with a training score which is pretty similar to the testing score.

2.4 Results

The best score we got is 0.86 for the Extra Trees Regressor after having applied a grid search on it. The representation below shows the order of the models's scores. But after applied the grid search we got a 0.86 score.

	Score R2
model	
Extra Trees Regressor	0.828773
Random Forest Regressor	0.819044
Decision Tree Regressor	0.702782
Linear regression	0.472014
Lasso regression	0.462381
Ridge regression	0.462287
SVR	0.373138

Figure 1: Plot presenting the models R2 scores

2.5 Discussion

Let's **discuss** about these algorithms's results to choose which one we will keep. The **metric** we have chosen is **the R2** in order to compare all the models we did. All the scores we compare are best estimators grid search scores. The algorithm that we have to keep is the algorithm which presents **the best test score**. So that when we deploy our machine learning API we have the best prediction following the features we put on the model. The model with the best score is **the Extra Trees Regressor**. That's why we decided to **deploy** this model into an API.

2.6 Transform the model to an API Flask

This is the final step of our project, the API. The model deployment is a very important step because it is what the client or the user will see of your work. The back end of the API will be the model implemented and the front end will be the interface in which you will input the values requested. The values requested are the features, the output will be the estimated target. We did a request forms for to get the features needed. The display is as well simple with a predict button to launch the prediction once having input variables. And we finally have the prediction printed below the Predict button.

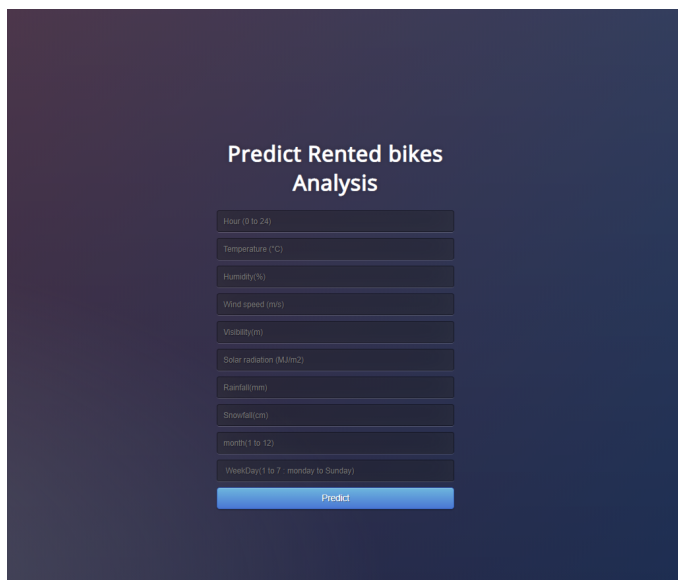


Figure 2: Image of the final API



3 Acknowledgment

This project was the opportunity for us to work on a concrete project. It was very pleasant for us to do it because we both like South Korea and Seoul and we were initially both drawn for this project, so we were both lucky. At the end of this project, we would like to thank Luc Bertin, our teacher who taught us how to manage a data analysis and gave good tips.

4 Conclusion

To conclude, we are very satisfied of the project we have done, because we reached our goals. This project taught and helped us understanding a lot about the use of machine learning models. It was the opportunity for us to manage the cycle of a machine learning model from the preprocessing to the deployment was very enriching. Indeed, this project was very enriching to do for us, because it permitted us to wonder real questions which could be transposed to our city : Paris. This study is a real study which big companies could ask and pay for so it helped us understand the need behind the study. We also have to highlight the fact that this work stimulated our thoughts because we had a lot of freedom on this project so that we could be creative.