

# Facebook Comment Volume Prediction Dataset



**Predict the number of comments for a Facebook  
post for the  $X$  next hours**



# Steps

① **Presentation of Dataset**

② **Inspection/Cleaning of  
data**

③ **Visualization of  
data**

④ **Machine Learning**

# Dataset

**Comments picked on several Facebook pages**

**Data are pre-treated:**

- > no comments older than 3 days before de data collect**
- > no incomplete data**

**Dataset split into 5 files corresponding to the number of data collecting base time.**

# Base Time

**Data are collected based on a X multiplier (from 1 to 5).**

**For each variant, we find X instances of each post with different base times.**

**Since the number of final comment is known, the author wanted to add difficulty, by adding the 'hrs' variable in my code, corresponding to a random integer between 1 and 24. This indicates the time gap between the base time and the prediction.**

# Inspection of data

Each variant possesses 53 numerical variables and 1 target.

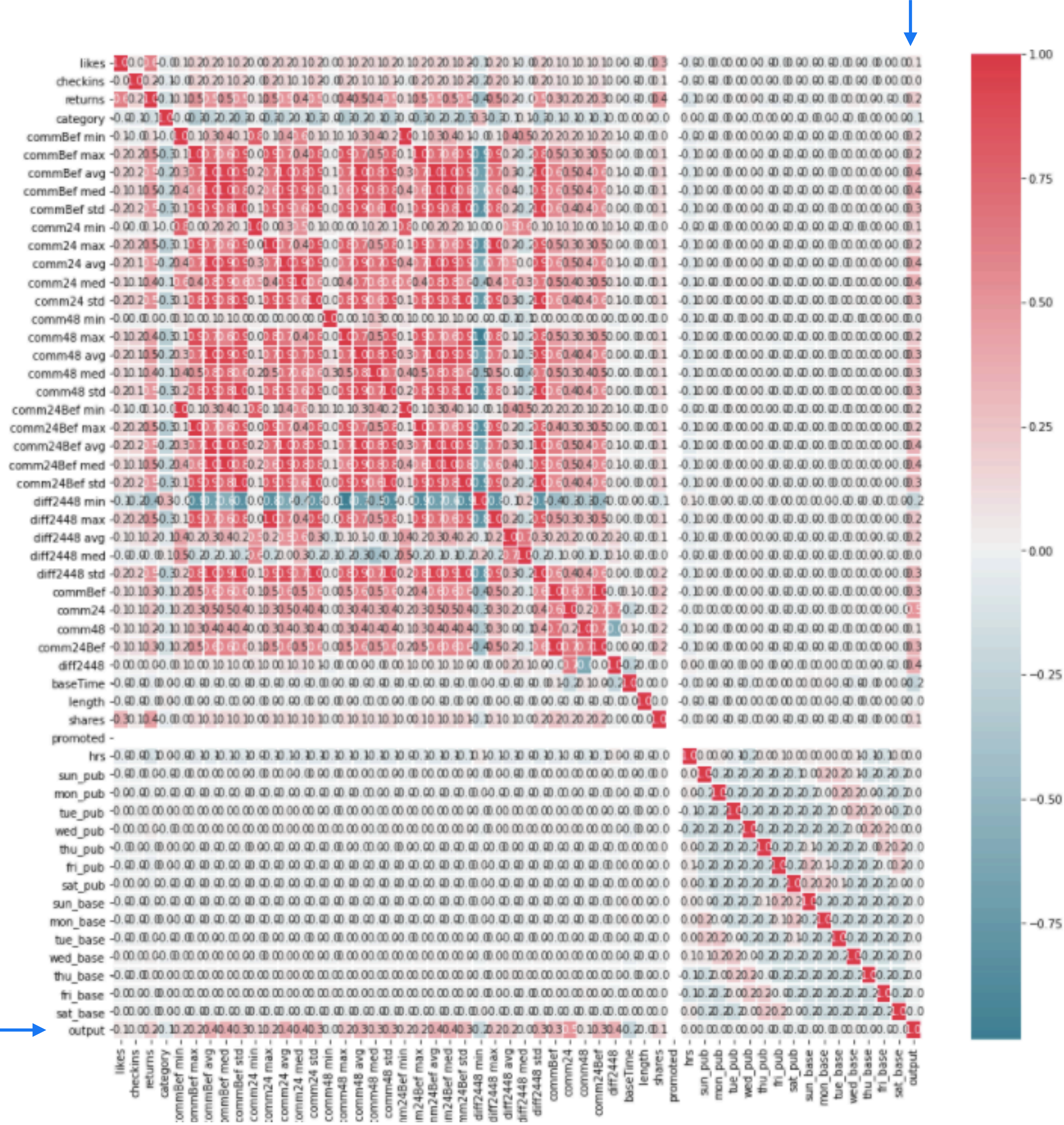
diff2448 max	diff2448 avg	diff2448 std	commBef	comm24	comm24Bef	diff2448	baseTime	promoted	hrs	sun_pub	mon_pub	tue_pub	wed_pub	thu_pub	fri_pub	sat_pub
806.0	4.970149	69.85058	0	0	0	0	65	0	24	0	0	0	1	0	0	0
806.0	4.970149	69.85058	0	0	0	0	10	0	24	0	0	0	0	1	0	0
806.0	4.970149	69.85058	0	0	0	0	14	0	24	0	0	0	0	0	1	0
806.0	4.970149	69.85058	7	0	7	-3	62	0	24	0	0	0	0	0	1	0
806.0	4.970149	69.85058	1	0	1	0	58	0	24	0	1	0	0	0	0	0
806.0	4.970149	69.85058	0	0	0	0	60	0	24	0	0	1	0	0	0	0
806.0	4.970149	69.85058	0	0	0	0	68	0	24	0	0	0	1	0	0	0
806.0	4.970149	69.85058	1	0	1	-1	32	0	24	0	0	0	0	1	0	0
806.0	4.970149	69.85058	0	0	0	0	35	0	24	0	0	0	0	0	1	0
806.0	4.970149	69.85058	0	0	0	0	48	0	24	0	0	0	0	0	1	0



**Here is an overview of the correlation matrix obtained.**

**The output target is resented on the last row and the last column.**

**We easily see thanks to the colour scale that the output is not very correlated with other variables.**





# Data cleaning

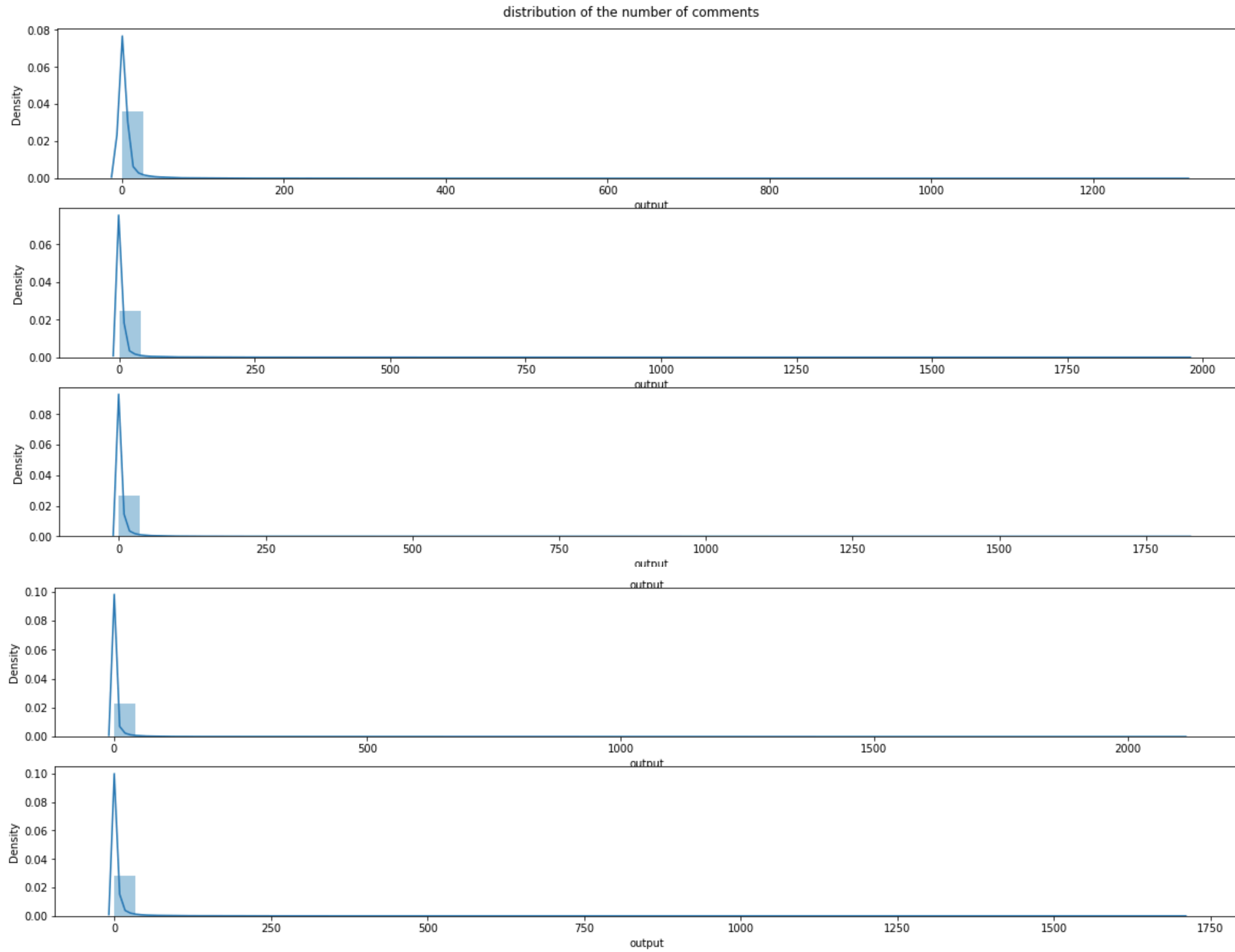
Cleaning of less influential parameters (correlation  $\in [-1.9 ; 1.9]$  )

Withdrawn parameters : 'checkins', 'comm24 min', 'comm24Bef min', 'comm48',  
'comm48 min', 'commBef min', 'diff2448 med', 'length', 'likes', 'returns', 'shares'

—> 53 => 42 parameters

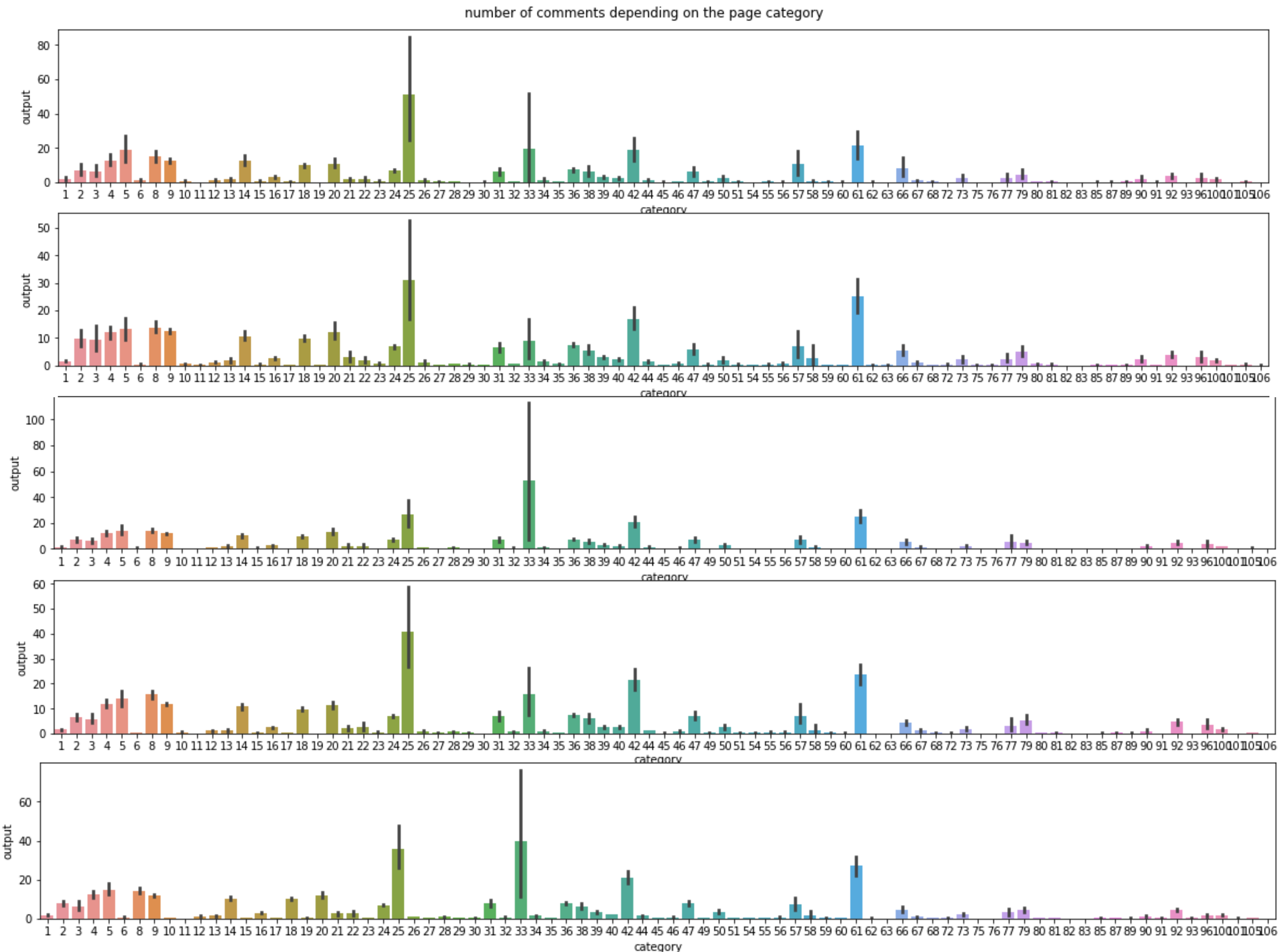


# Data Visualization



**First, let's see the distribution of the target value.**

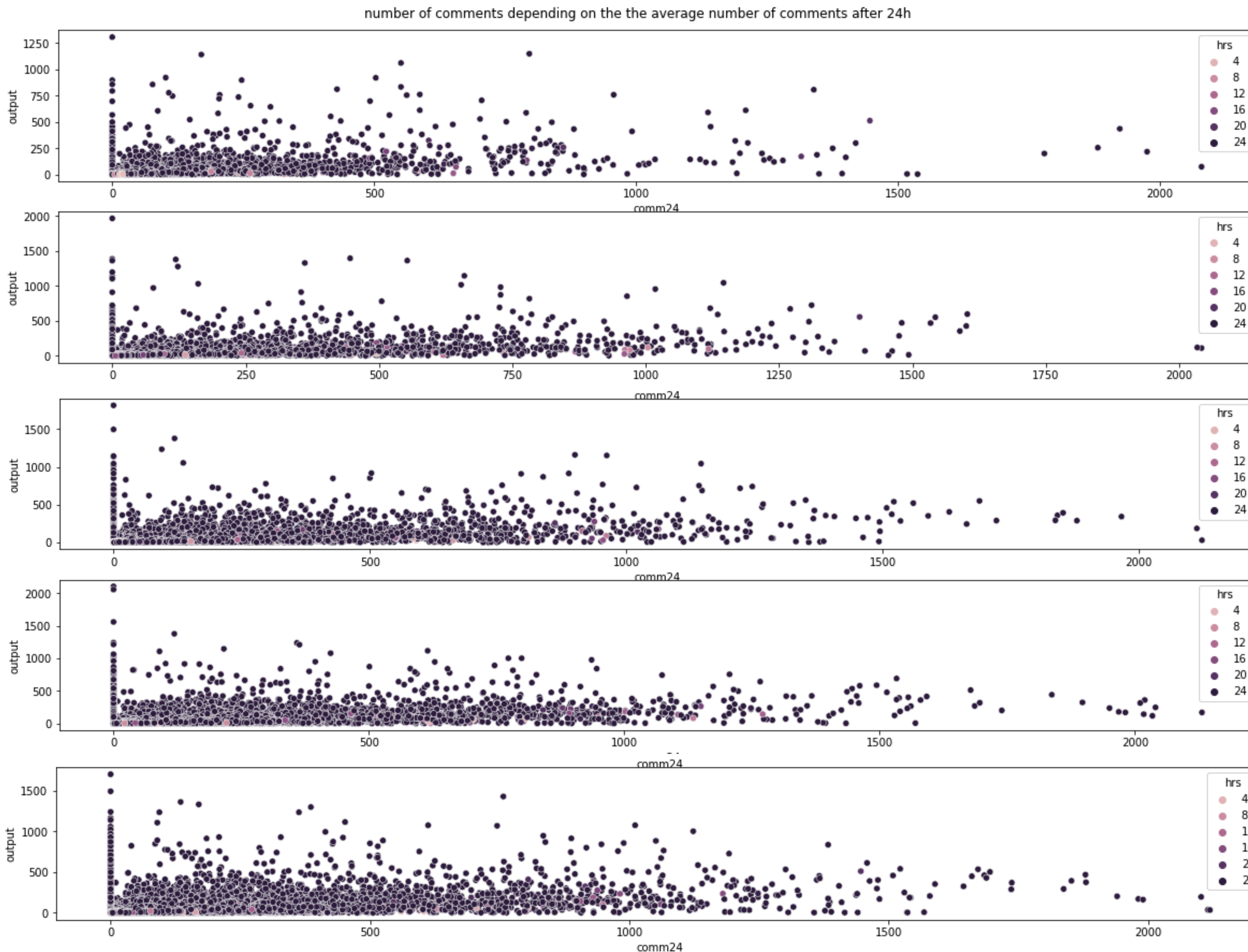
**As we can see, most of the data are null or very low (-100)**



Here, we find the number of comments depending on the page category.

We still notice that some categories stand out. That's why I decided to keep this parameter for the predictions.





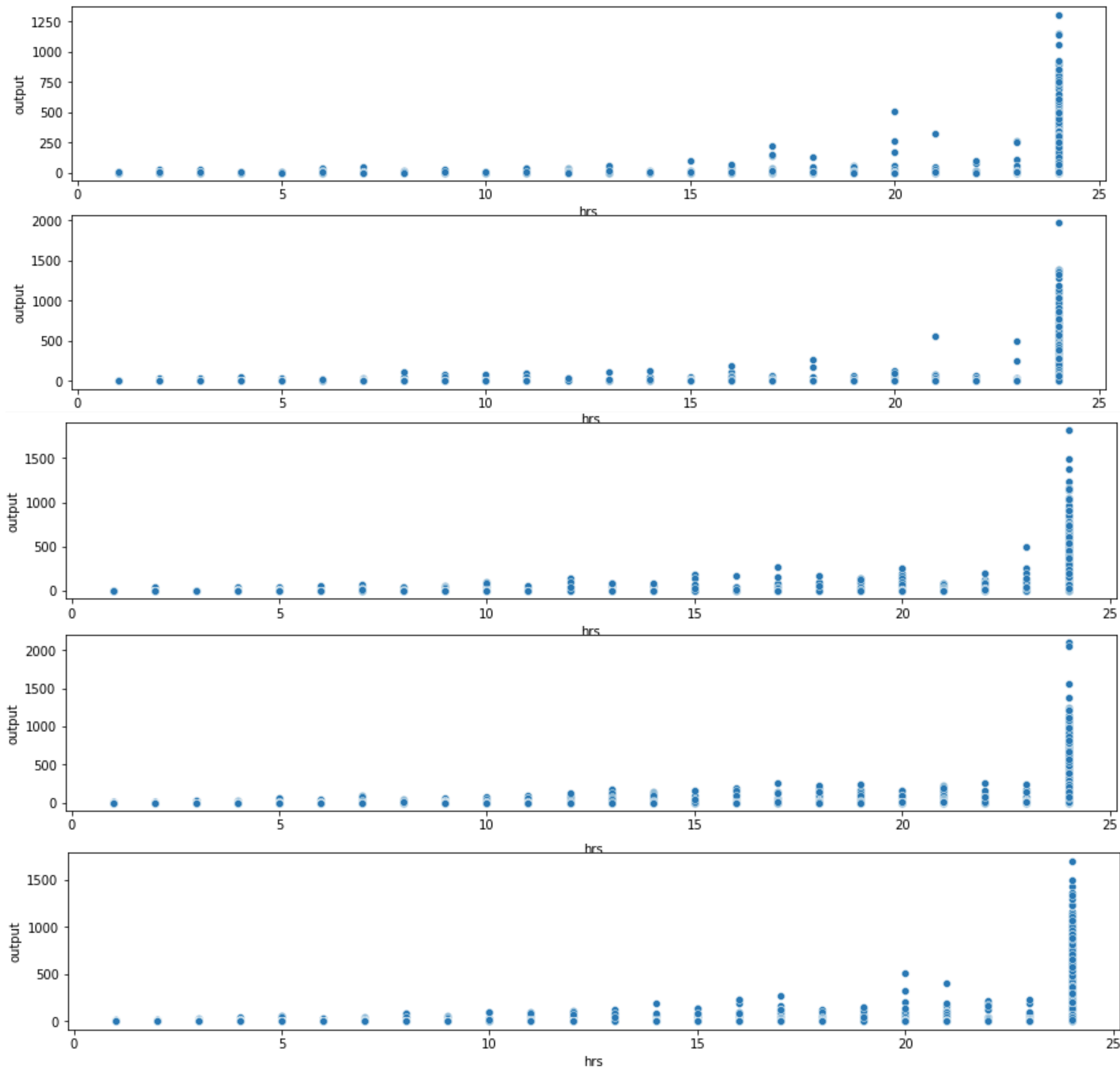
**Here, we find the number of comments depending on the number of comments already present 24h before the associated base time.**

**I added the hour gap between the base time and the prediction with the hue parameter.**

**Most of the prediction are made 24h after the base time. Even if it is the most influential variable, we can't seem to see the link.**



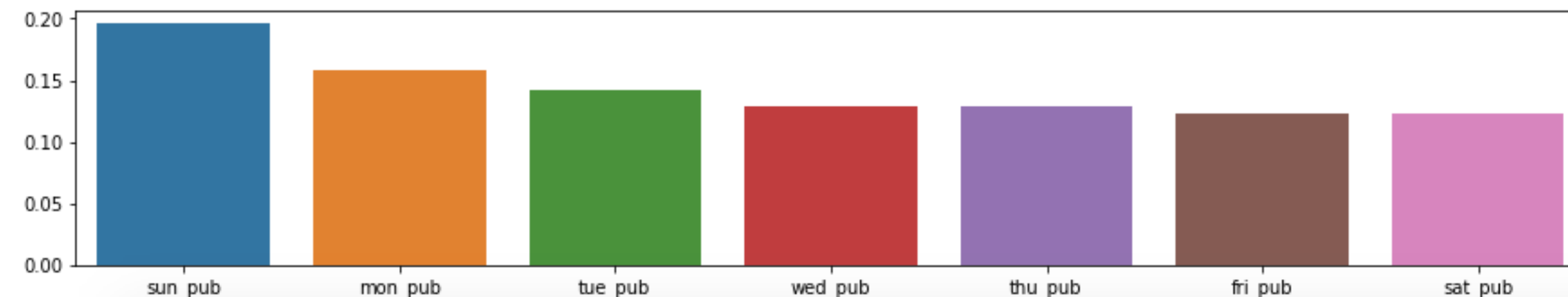
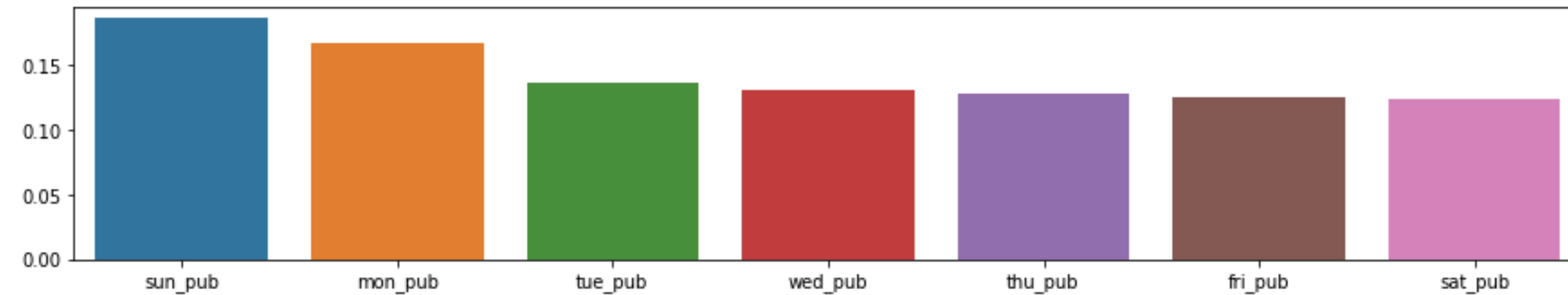
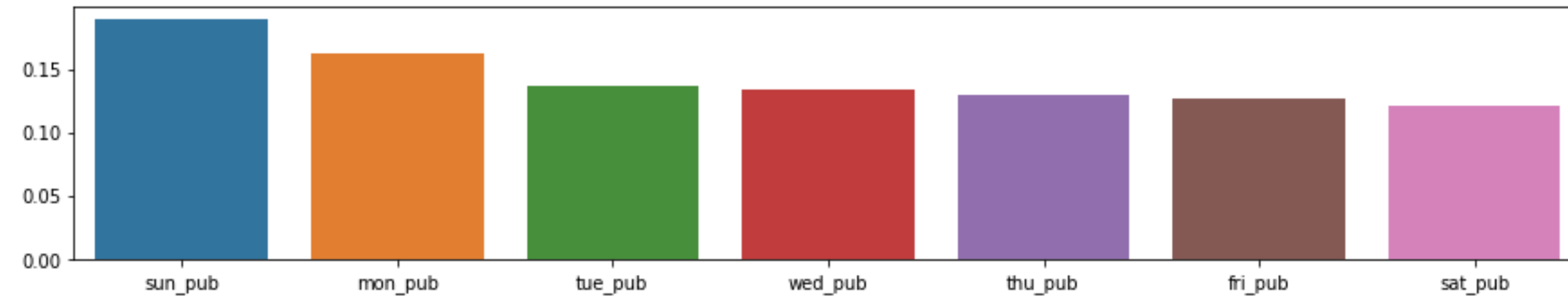
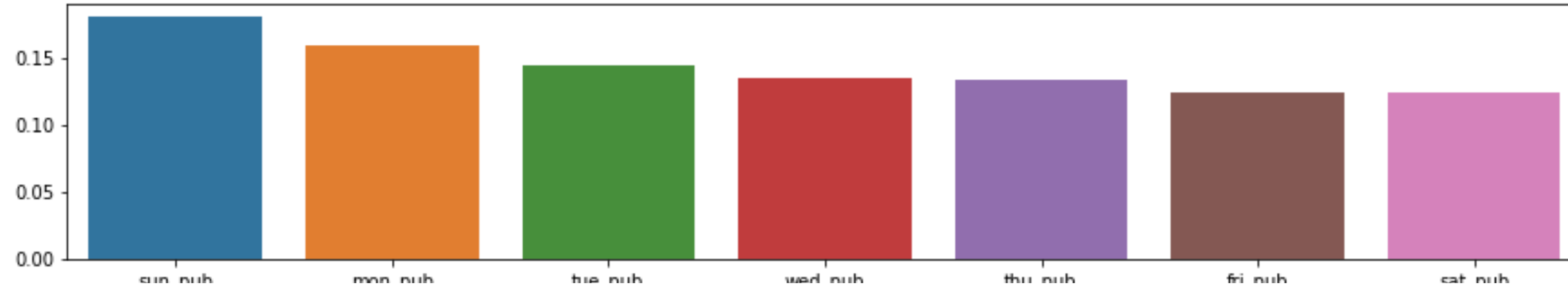
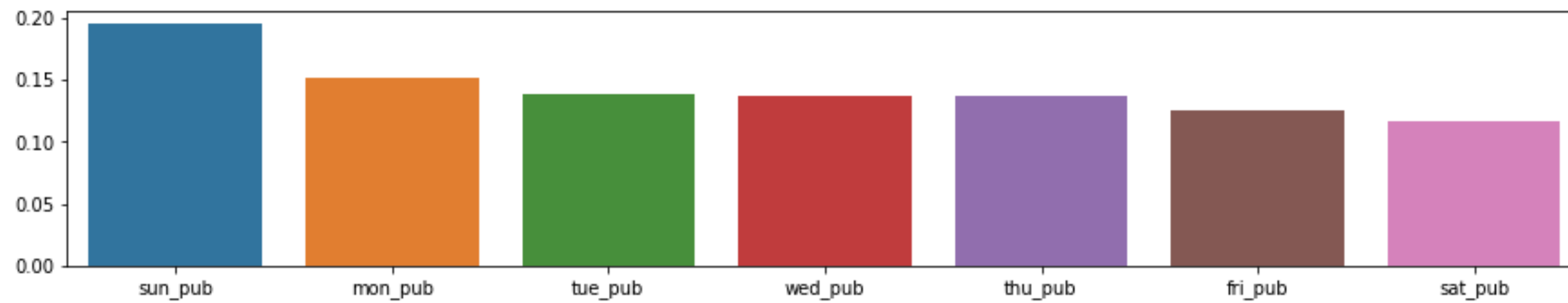
number of comments depending on the next hours prediction



**To confirm the tendency of predictions mostly asked 24h after the base time, I plotted the graphics depending on the variable 'hrs'.**

**As we suspect, post receive more comments during a longer time period. Moreover, it confirms that 24h is the most common period within the dataset.**

ratio of comments depending on the publication date

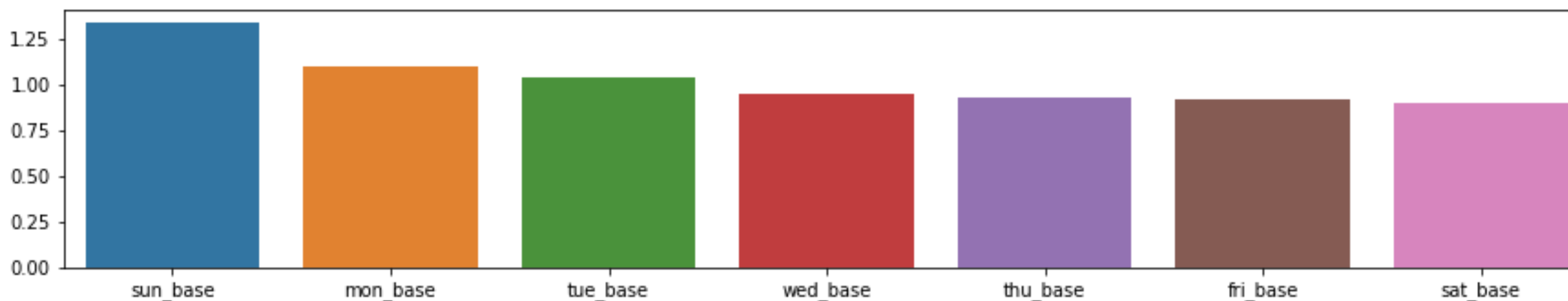
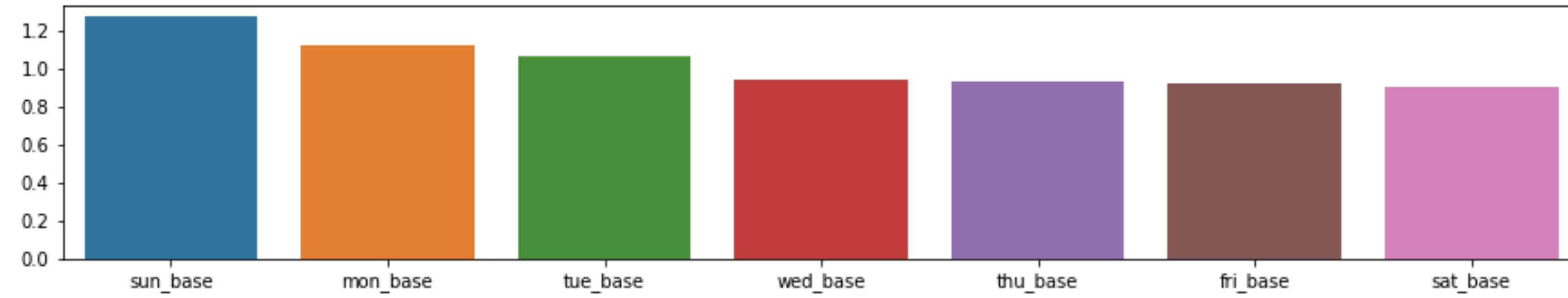
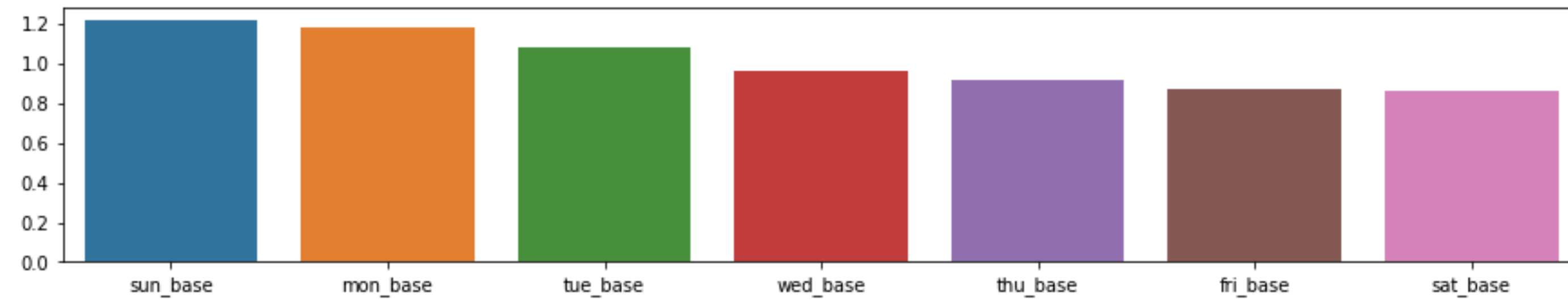
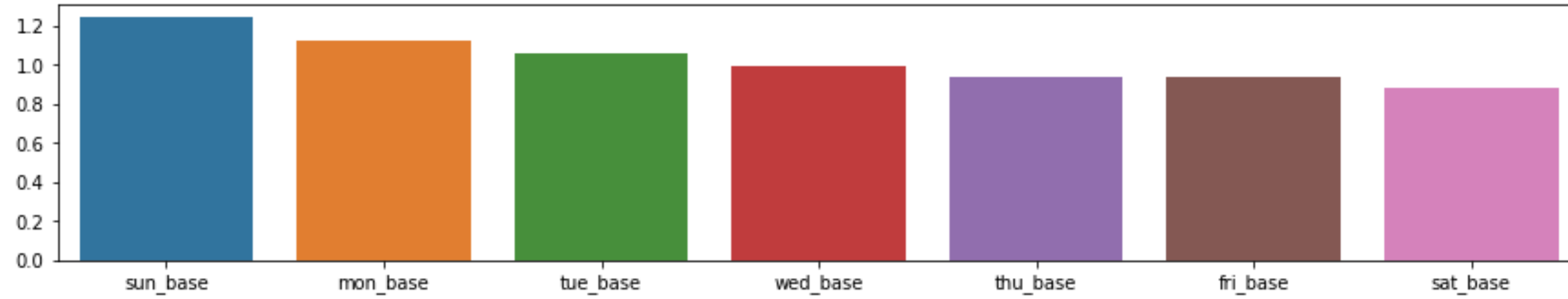
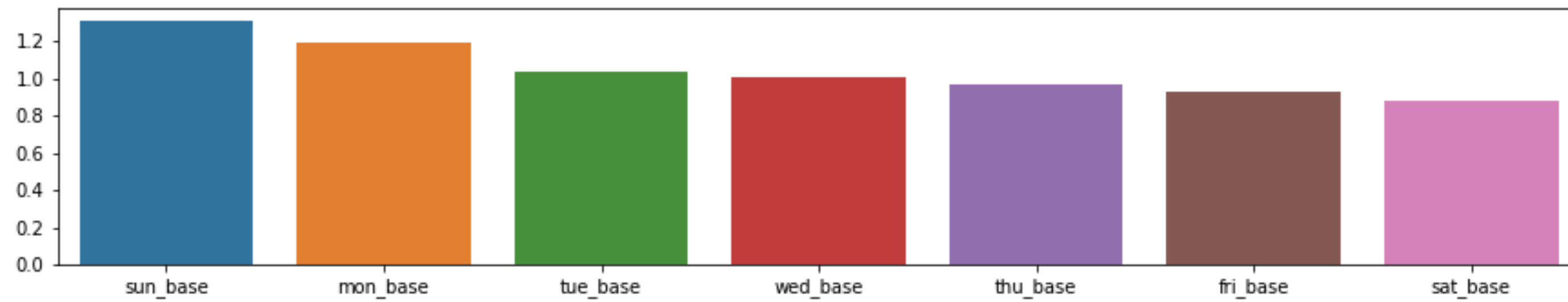


The histogram opposite represents the days during which the various posts were published.

To have a more precise idea, I made a ratio of the number of comments compared to the total number of comments.

You can notice that it's for those published on a Sunday, and generally at the beginning of the week that we have the most comments.

ratio of comments depending on the baseTime date



**We find the same pattern for the day of the baseTime chosen to collect the data.**

**These results are not surprising because comments are predicted for a maximum of 24 hours after publication.**



# Machine Learning

## Selected models :

- Linear Regression
- Random Forest
- Decision Tree
- Elastic Net

## Best Parameters :

- Linear Regression :

`'fit_intercept' : False, 'normalize' : True`

- Random Forest :

`'max_depth' : 10, 'max_features' : 'auto', 'n_estimators' : 50`

- Decision Tree :

`'max_depth' : 6, 'criterion' : 'friedman_mse', 'max_features' : 'auto'`

- Elastic Net :

`'alpha' : 6, 'l1_ratio' : 6`

Optimization of hyper parameters using a search grid for all models.

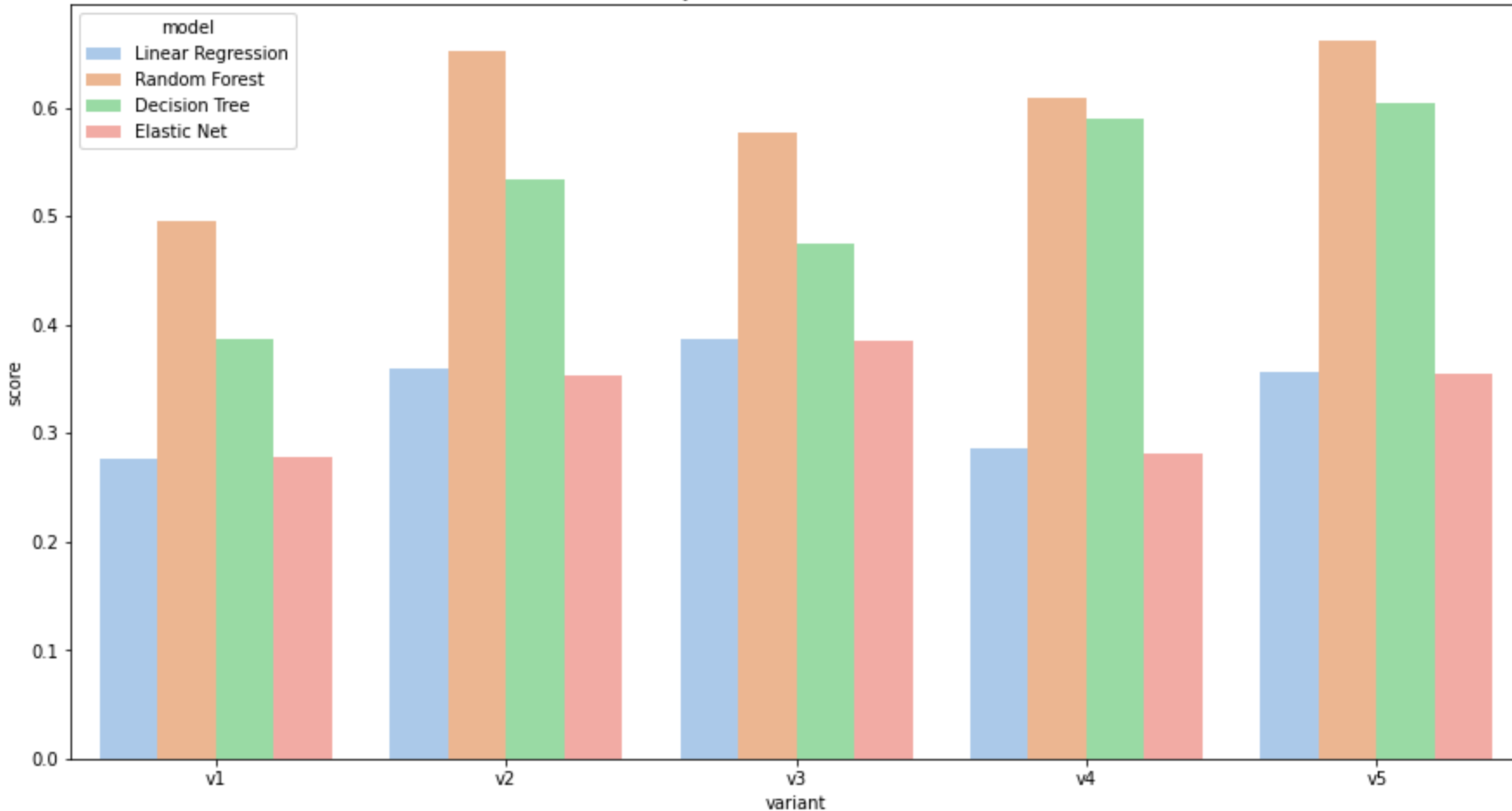
# Results of the models

	variant	model	score
0	v1	Linear Regression	0.275785
1	v2	Linear Regression	0.360291
2	v3	Linear Regression	0.386607
3	v4	Linear Regression	0.285650
4	v5	Linear Regression	0.355495
5	v1	Random Forest	0.495624
6	v2	Random Forest	0.652346
7	v3	Random Forest	0.577130
8	v4	Random Forest	0.609556

	variant	model	score
14	v5	Decision Tree	0.603665
15	v1	Elastic Net	0.277286
16	v2	Elastic Net	0.353298
17	v3	Elastic Net	0.384892
18	v4	Elastic Net	0.281827
19	v5	Elastic Net	0.354332

# Comparison of the models

Comparison of models



**For all our variants, the most efficient model is the Random Forest (accuracy between 49% and 66%).**

**followed by the Decision Tree (accuracy between 38% and 60%)**

**Finally, Linear Regression and Elastic Net have similar results (accuracy between 27% and 38%).**



# API with Flask

For the API, I decided to save the models fit with the 5th variant since it's the variant with the most accuracy since it was trained with more data.

Moreover, all variant have the same parameters so it was useless to use all of them.

[Home](#) | [Make prediction](#)

## Welcome to the Predict Comment Volume of a Facebook Post API

You can predict the number of comment that a post will get based on 2 characteristics :

The first is the Base Time ie how long between the publication time and the collect time.

The second is the time gap between the base time and the prediction.

For our models, we use the fitted model with the 5th variant (each instance is multiplicated 5 times)

In the prediction page, you will have the possibility to change every of the 42 parameters.

If you don't want to fill every input, don't worry ! Each time you click on the 'Make Prediction' button, new data will be automatically filled.

Start

# API with Flask

Here the user can change the pre-filled data

[Home](#) | [Make prediction](#)

## Make Prediction

Regression Linear

Random Forest

Decision Tree

Elastic Net

Category (1 to 107) <input type="text" value="37,0"/>	Comments Before max <input type="text" value="2121,0"/>	Comments Before avg <input type="text" value="683,0"/>
Comments Before med <input type="text" value="527,5"/>	Comments Before std <input type="text" value="636,0991918075214"/>	Comments 24h max <input type="text" value="1873,0"/>
Comments 24h avg <input type="text" value="496,5"/>	Comments 24h med <input type="text" value="214,0"/>	Comments 24h std <input type="text" value="569,3761937418881"/>
Comments 48h max <input type="text" value="1265,0"/>	Comments 48h avg <input type="text" value="154,54545454545453"/>	Comments 48h med <input type="text" value="0,0"/>
Comments 48h std <input type="text" value="331,03799936683015"/>	Comments 24h Before max <input type="text" value="1937,0"/>	Comments 24h Before avg <input type="text" value="670,7727272727273"/>
Comments 24h Before med <input type="text" value="522,5"/>	Comments 24h Before std <input type="text" value="617,6686322424009"/>	Comments between 48h and 24h min <input type="text" value="-1112,0"/>
Comments between 48h and 24h max <input type="text" value="1873,0"/>	Comments between 48h and 24h avg <input type="text" value="341,95454545454544"/>	Comments between 48h and 24h std <input type="text" value="730,3334529373133"/>
Comments Before <input type="text" value="0,0"/>	Comments 24h <input type="text" value="0,0"/>	Comments 24h Before <input type="text" value="0,0"/>
Comments between 48h and 24h <input type="text" value="0,0"/>	Base Time <input type="text" value="0,0"/>	Promoted <input type="text" value="0,0"/>
Gap between the Base Time and the Prediction <input type="text" value="9,0"/>	Day of Publication (1:sunday 7:saturday) <input type="text" value="1,0"/>	
Day of Collect (1:sunday 7:saturday) <input type="text" value="1,0"/>		

# API with Flask

Home | Make prediction

## Decision Tree

Number of comments predicted : 488.84

Regression Linear

Random Forest

Decision Tree

Elastic Net

To make a prediction with another model and the same values, click on one of the button above.

To make a new prediction with different values, click on the button 'Make prediction' in the header.

Home | Make prediction

## Random Forest

Number of comments predicted : 774.48

Regression Linear

Random Forest

Decision Tree

Elastic Net

To make a prediction with another model and the same values, click on one of the button above.

To make a new prediction with different values, click on the button 'Make prediction' in the header.

We can see the different results obtained with all our models with the same values

Home | Make prediction

## Linear Regression

Number of comments predicted : 75.61

Regression Linear

Random Forest

Decision Tree

Elastic Net

To make a prediction with another model and the same values, click on one of the button above.

To make a new prediction with different values, click on the button 'Make prediction' in the header.

Home | Make prediction

## Elastic Net

Number of comments predicted : 73.96

Regression Linear

Random Forest

Decision Tree

Elastic Net

To make a prediction with another model and the same values, click on one of the button above.

To make a new prediction with different values, click on the button 'Make prediction' in the header.