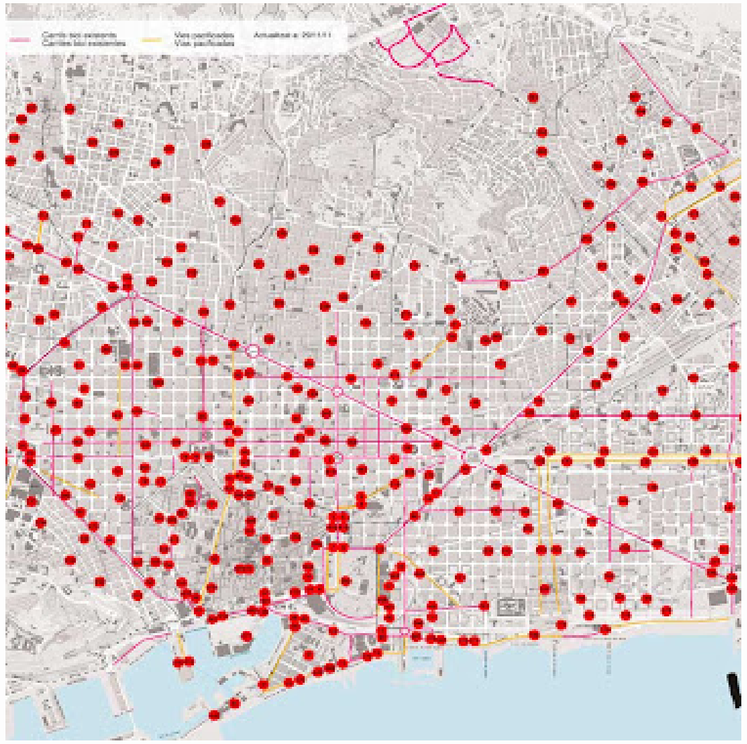
MOBILITY MAINTENANCE

JUAN CARLOS RUBIO GIL

A MACHINE LEARNING MODEL

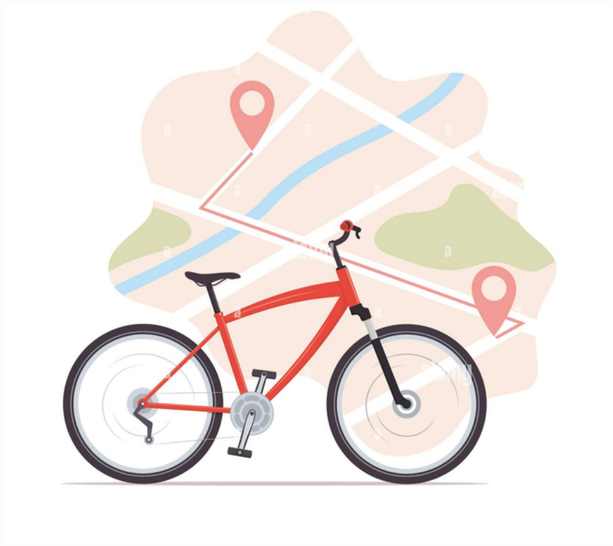


OBJETIVE & CIRCUSTANCES

The scenario presented involves a bike rental company in a city. The problem to be addressed coexists with the need for bike maintenance. Thus, the model aims to answer how much usage load the bike system will have to be able to take out of circulation those bikes that need reviews or maintenance without affecting the service.

That is, to predict the number of people who will use the service, at what times, on which days, and which parameters influence this in order to determine the best time to withdraw the vehicles.

DATA



To achieve this, the model uses the database that can be found in one of Kaggle's competitions, about the bike service in Washington DC. Among the data are the date and time and data related to it, the weather conditions, and the number of people who used the service in each time slot. Here are the columns broken down:

- datetime - hourly date + timestamp
- season - 1 = spring, 2 = summer, 3 = autumn, 4 = winter
- holiday - whether the day is considered a holiday
- workingday - whether the day is neither a weekend nor holiday
- weather:
  - 1: Clear, Few clouds, Partly cloudy, Partly cloudy
  - 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
  - 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds
  - 4: Heavy Rain + Ice Pellets + Thunderstorm + Mist, Snow + Fog
- temp - temperature in Celsius
- atemp - "feels like" temperature in Celsius
- humidity - relative humidity
- windspeed - wind speed
- casual - number of non-registered user rentals initiated
- registered - number of registered user rentals initiated
- count - total number of rentals

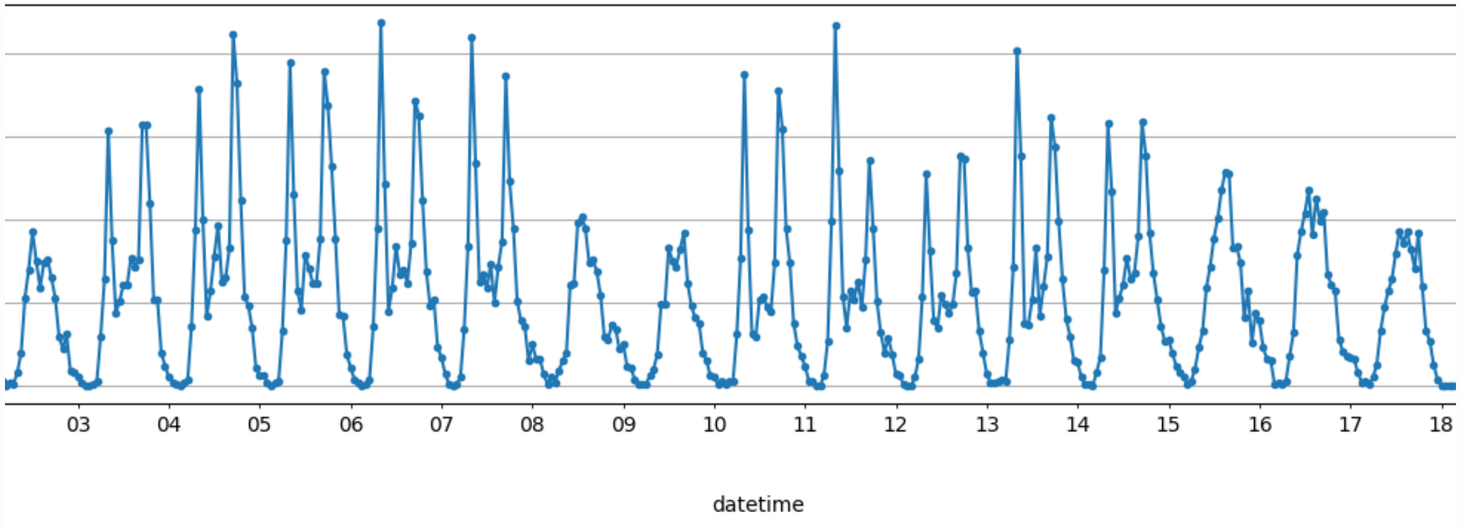
HOW

01 The adopted solution involves converting the dates into a format that functions as columns with information such as hour, month, day of the week... instead of treating it as a time series.

02 Subsequently, a regression model is applied to obtain the number of people who will be using the service at each moment.

The idea is that with the weather forecast and the days of the week, we are able to predict the use of the service.

E D A

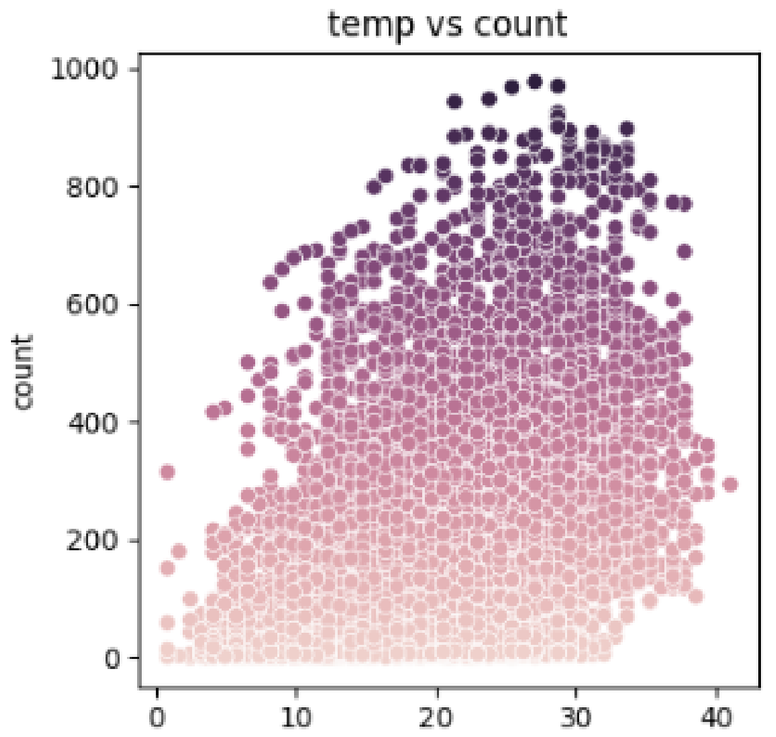


The image shows the distribution of the target "count" throughout the hours over a month.

Note that the data we have are from the first 19 days of each month of the years 2011 and 2012. This will already establish certain limitations for making time series.

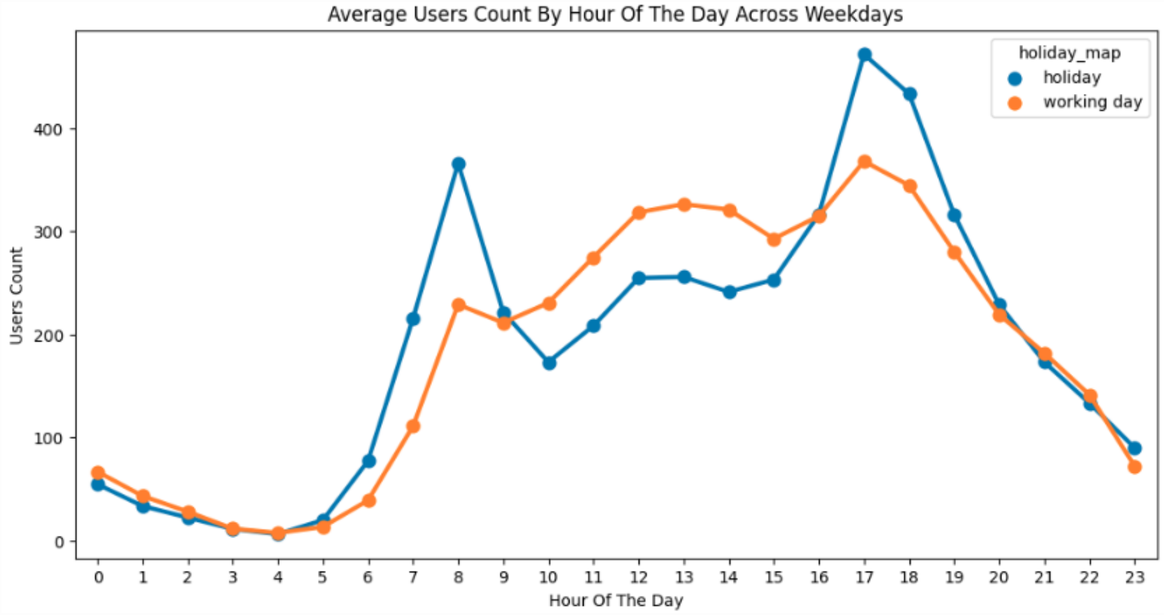
Analyzing the image, we can observe a similar rhythm in the period of a day as well as the weekly period, though with differences between the days. It is not constant. This leads us to think that other elements have a decisive influence on the use of this means of transportation. In this case, we assume that it is the weather or whether the day is a working day or not, and we try to set up a prediction model with those data, to see how it goes.

EDA



01 Here we can see how the use of the bicycle varies according to temperature; people prefer higher temperatures over cold ones, but usage also drops if it's too high.

02 Here we can see how whether it is a working day or not varies the use that is made of the bicycle during the different hours of the day.



DATA MANAGEMENT

01 We extract the date, time, year, month, and day of the week from datetime into their own columns.

02 We apply a logarithmic transformation to windspeed to normalize it.

03 We drop the columns: datetime, year, date, atemp and season.

04 Escale with MinMaxScaler()

05 We perform One_hot_encoding on hour, month y weekday

MODEL HISTORY

Baseline

Treating it as a time series, we test a 7-day naive model and take it as a baseline. Autoarima yields worse results, so we discard it.

First steps

Initially used are: 'Ridge', 'Lasso', 'ElasticNet', 'SVR', 'DecisionTreeRegressor', 'RandomForestRegressor', 'GradientBoostingRegressor', 'XGBRegressor', 'LGBMRegressor', and 'CatBoostRegressor'.

Sticking with 'GradientBoostingRegressor', 'XGBRegressor', and 'LGBMRegressor' in the first filter.

Optimization

Hyperparameters are optimized with cross-validation and against the validation dataset.

Final Training

The train and validation sets are combined to train the new model, and it is then used to make predictions, in this case, on the test set provided by Kaggle. Upon checking, we obtain a relatively small error.

We use the RMSLE metric because RMSLE is employed to minimize the impact of large errors in predictions, making it ideal when it's more important to avoid significant underestimations than overestimations.

MODEL

	MAE	MAE_Diff_Baseline	MAPE	MAPE_Diff_Baseline	RMSLE	RMSLE_Diff_Baseline
LinearRegression	79.6674045077	75.2870514984	3395106.3892800175	3395052.9309146334	1.0696860120	0.3360499756
Ridge	79.6607244739	75.2803714646	3365036.4965403052	3364983.0381749212	1.0643051042	0.3306690678
Lasso	79.7005965983	75.3202435890	3181922.2412731680	3181868.7829077840	1.0524771506	0.3188411143
ElasticNet	100.3937745155	96.0134215063	3540599.9687087731	3540546.5103433891	1.2656639097	0.5320278734
SVR	111.5946497769	107.2142967676	3202890.0564282350	3202836.5980628510	1.2376321654	0.5039961290
DecisionTreeRegressor	101.2496203333	96.8692673241	4409156.4788486855	4409103.0204833020	1.3321558698	0.5985198334
RandomForestRegressor	96.5805687848	92.2002157756	4384934.2667746665	4384880.8084092829	1.3186120986	0.5849760622
GradientBoostingRegressor	66.8606984407	62.4803454315	1709610.0498339017	1709556.5914685179	0.8556541200	0.1220180836
XGBRegressor	51.5534775048	47.1731244955	808234.2426200376	808180.7842546537	0.6005609618	-0.1330750745
LGBMRegressor	51.4925784973	47.1122254881	1090371.5909839855	1090318.1326186017	0.7576572694	0.0240212330
CatBoostRegressor	73.1297995710	68.7494465617	1660794.7877604750	1660741.3293950912	0.9305668956	0.1969308592

01

Here we can se the scores in the first appoarch.

02

Here we have the val scores from selected models

and the best score versus test which is GradientBoostingRegressor

```
XGBRegressor: 0.5063773397093668
GradientBoostingRegressor: 0.5233330994310037
LGBMRegressor: 0.5921671664151393
```

TO SUM UP

The model manages to predict the number of bicycles that will be used with considerable accuracy, making it a viable solution to the considered business problem. Moreover, it is particularly effective, thanks to the chosen metric, at not underestimating demand to prevent a shortfall in service.

ENHANCEMENT

Events

Looking forward to enhancing the model, it would be interesting to introduce variables such as specific events to see if and how they influence the use of the service.

Location

It would be interesting to make spatially localized predictions, as it's likely that the demand for the service varies greatly by area.

In the long run

The model is unable to account for trends. Probably, the days closer in time should weigh more than those further away.

submission_lgbmr.csv Complete (after deadline) · now · second lbgmr model	**0.65889**	**0.65889**	☐
submission_gbr.csv Complete (after deadline) · 19s ago · gbr model	**0.61100**	**0.61100**	☐
submission_xgbr.csv Complete (after deadline) · 35s ago · xgbr model	**0.62813**	**0.62813**	☐
submission_lgbmr.csv Complete (after deadline) · 7h ago · Submission from a LGBMRegressor model	**0.62760**	**0.62760**	☐
submission_dtr.csv Complete (after deadline) · 7h ago · Submission from a DecisionTreeRegressor model	**0.55328**	**0.55328**	☐
submission_rfr.csv Complete (after deadline) · 7h ago · Submission from a RandomForestRegressor model	**0.51816**	**0.51816**	☐