

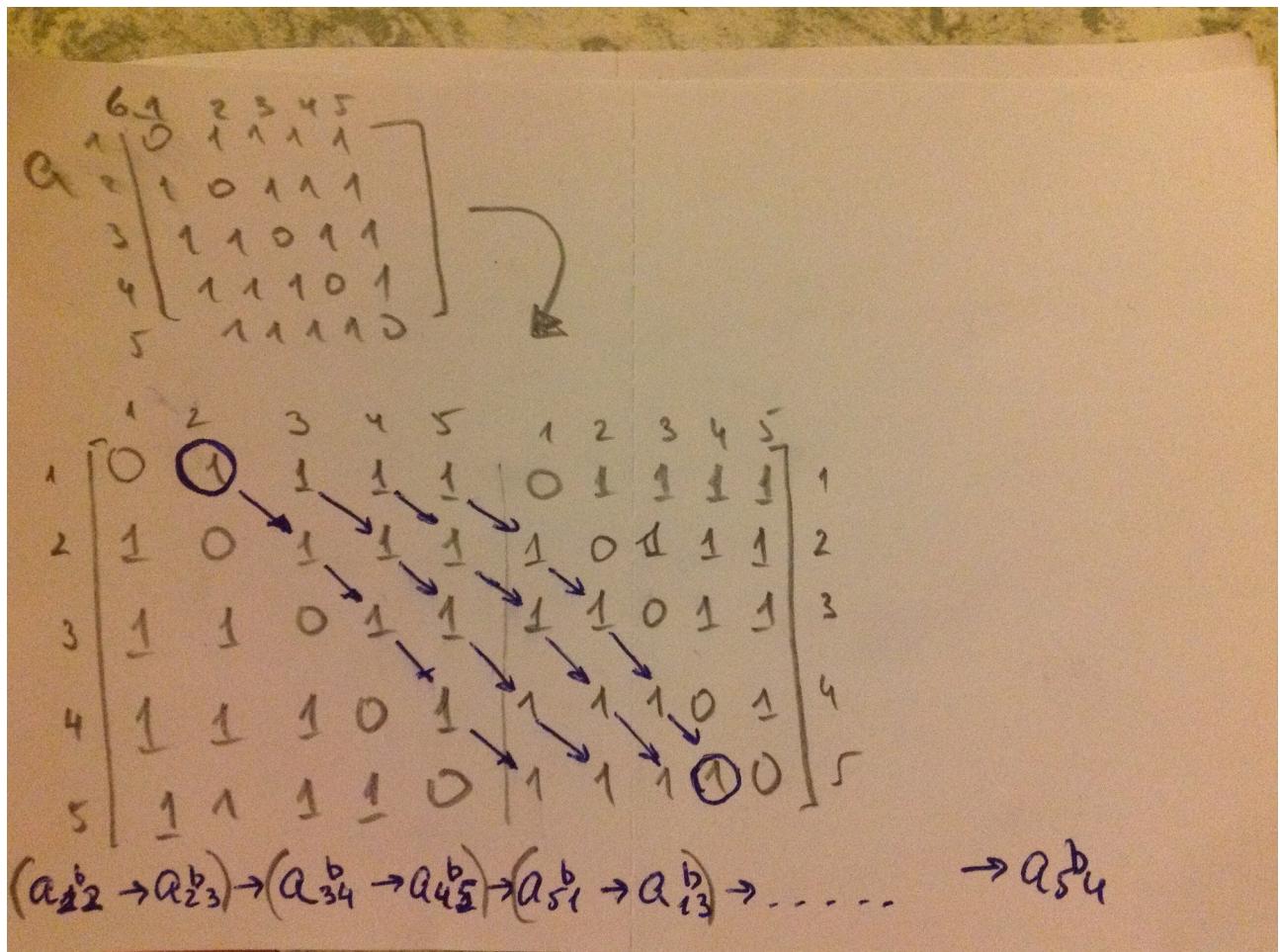
Task 1

I'm not sure that I understood task right, because description is pretty obscure.

So, first of all I've written all required pairs in matrix style (top of the picture), then chosen elements diagonal traverse way (arrows at picture), then grouped them in pairs.

The schedule ends when all non-zero elements are passed.

To optimize at time I just assumed that one match takes 1 quant of time and realized that pair formation strikes out corresponding columns and rows. So, I incremented time when I was out of free columns/rows or out of available courts.



Task 2

Due to it's not a Kaggle competition and rigorous restrictions are not presented, I've stopped my solution when dummy benchmark was beaten.

Solution consists of three parts:

1. Short data research
2. Data preprocessing
3. Model training

First part is uninteresting in my solution. During second part I've made couple assumptions:

1. I've work with data like with unique records, so I dropped all id fields
2. Converted timestamps to hours and day-of-week (because I think it makes sense)
3. Log transformed some time features and added flag to 0 values
4. Reduced variations of rare categorial values – merged them to “undefined” value
5. One-hot-encoded all categorial features

Third part is just a training of several models and evaluation their performance with respect to dummy benchmark (use mode of target variable = all 0).

Other things than can be done if it was production problem:

- NLP tools for text data
- More clever merge of rare data
- NA imputation
- Appropriate use of latitude and longitude (convert them to sectors, for.ex.)
- Model parameters optimization, stacking, bagging, averaging
- Feature selection based on their importance
- Use time-series nature of data for more accurate prediction
- and a lot of more cool stuff