

Babar Forecasting

2020

Chapter 1

Goal

1.1 The Problem

1.2 Our Solution

Chapter 2

Data Management

2.1 Selection and Gathering Sales Data

There are too many products to have a prediction for each one. So we decided to gather them into categories. We want to gather them into pertinent groups so that the forecast can really help the bar.

2.1.1 First Selection

The first step is to choose which product we are going to take account of. We choosed to focus our work on the drinks sales that had at least been sold ???. So we exclude any food or other kind of sales.

SQL Request too extract those product :

```
CREATE TABLE WantedProducts AS SELECT babar_server_purchase.* FROM babar_server_purchase
JOIN babar_server_product ON product_id = babar_server_product.id WHERE NAME = 'Leffe' OR
NAME = 'Hoegaarden blanche' OR NAME = 'Desperados' OR NAME = 'Smirnoff' OR NAME =
'Pastis' OR NAME = 'Hard' OR NAME = 'Grimbergen' OR NAME = 'Chimay Rouge' OR NAME
= 'Chimay Bleue' OR NAME = 'Kro Demi' OR NAME = 'Cidre Demi' OR NAME = 'Pelforth'
OR NAME = 'Kwak' OR NAME = 'Kir' OR NAME = 'Kro Pinte' OR NAME = 'Cidre Pinte'
OR NAME = 'Cocktail Hard' OR NAME = 'Chimay Blanche' OR NAME = 'Shot' OR NAME =
'Blanche Demi' OR NAME = 'Blanche Pinte' OR NAME = 'Cidre Doux/Brut' OR NAME = 'Am-
bree demi' OR NAME = 'Ambree Pinte' OR NAME = 'Delirium' OR NAME = 'Rouge Pinte' OR
NAME = 'Sangria' OR NAME = 'Karmeliet Triple' OR NAME = 'Duvel' OR NAME = 'Granita
Hard' OR NAME = 'Skoll' OR NAME = 'Rouge Demi' OR NAME = '1664 Blanche' OR NAME
= 'Chimay bleue' OR NAME = 'Pecheresse' OR NAME = 'Cuvee des Trolls' OR NAME = 'Kriek'
OR NAME = 'Elephant Pinte' OR NAME = 'Elephant Demi' OR NAME = 'Maredsous Triple' OR
NAME = 'Hard Qualite' OR NAME = 'BrewDog Punk IPA' OR NAME = 'JagerBomb' OR NAME
= 'Cubanisto' OR NAME = 'Chouffe Pinte' OR NAME = 'Chouffe Demi' OR NAME = 'Corona'
OR NAME = 'Tigre Bock' OR NAME = 'Troll Pinte' OR NAME = 'Troll Demi' OR NAME =
'Triple Karmeliet Pinte' OR NAME = 'Triple Karmeliet Demi' OR NAME = 'Paix Dieu 33cL' OR
NAME = 'Grim Triple Demi' OR NAME = 'Grim Triple Pinte' OR NAME = 'Cherry chouffe' OR
NAME = 'Delirium rouge pinte' OR NAME = 'bush ambree' OR NAME = 'San Miguel';
```

List of considered product :

So now we have a data table gathering only the product we choose.

2.1.2 Other products that could be considered

2.1.3 Regrouping them into categories

A majority of this product has been coming and leaving from the bar menu so their sales aren't usable one by one. Therefor we decided to group them into pertinent categories. But how to choose this categories ? Here are our first thought.

High Degree Beers

- Chimay Bleu
- Kwak
- Karmeliet Triple
- Duvel
- Chimay bleue
- Maredsous Triple
- Chouffe Pinte
- Chouffe Demi
- Triple Karmeliet Pinte
- Triple Karmeliet Demi
- Grim Triple Pinte
- Grim Triple Demi
- bush ambree
- Delirium
- Elephant Pinte
- Elephant Demi

Normal Degree Beers

- Leffe
- Grimbergen
- Kro Demi
- Kro Pinte
- Pelforth
- Skoll
- BrewDog Punk IPA
- Tigre Bock
- Troll Pinte
- Troll Demi
- Cuvée des trolls
- Paix Dieu 33cL
- San Miguel
- Ambrée Pinte
- Ambrée Demi

Not Beer	Special Beers	Aromatized Beer
<ul style="list-style-type: none"> • Smirnoff • Pastis • Hard • Kir • Cocktail Hard • Shot • Rouge Pinte • Rouge Demi • Sangria • Granita Hard • Hard Qualite • JagerBomb 	<ul style="list-style-type: none"> • Desperados • Cidre Demi • Cidre Pinte • Cidre Doux/Brut • Cubanisto • Corona 	<ul style="list-style-type: none"> • Chimay Rouge • Hoegaarden blanche • Chimay Blanche • Blanche Demi • Blanche Pinte • 1664 Blanche • Pecheresse • Kriek • Cherry chouffe • Delirium rouge pinte

Other idea : Aromatize beer and Special together

Those are ideas of the categories, at every time at least one product of each category was being sold, assuring the continuity of sales in each categories. The goal is having sales that are homogene on a year scale.

2.1.4 SQL code for product gathering

```
DROP TABLE IF EXISTS 'High_Degree_Beer';
```

```
CREATE TABLE 'High_Degree_Beer' AS SELECT babar_server_purchase.* FROM babar_server_purchase
JOIN babar_server_product ON babar_server_purchase.product_id = babar_server_product.id WHERE
NAME = 'Chimay Bleue' OR NAME = 'Chimay Bleu' OR NAME = 'Kwak' OR NAME =
'Karmeliet Triple' OR NAME = 'Duvel' OR NAME = 'Maredsous Triple' OR NAME = 'Chouffe
Pinte' OR NAME = 'Chouffe Demi' OR NAME = 'Triple Karmeliet Pinte' OR NAME = 'Triple
Karmeliet Demi' OR NAME = 'Grim Triple Pinte' OR NAME = 'Grim Triple Demi' OR NAME
= 'bush ambrée' OR NAME = 'Delirium' OR NAME = 'Elephant Pinte' OR NAME = 'Elephant
Demi';
```

```
DROP TABLE IF EXISTS 'Normal_Degree_Beer';
```

```
CREATE TABLE 'High_Degree_Beer' AS SELECT babar_server_purchase.* FROM babar_server_purchase
JOIN babar_server_product ON babar_server_purchase.product_id = babar_server_product.id WHERE
NAME = 'Leffe' OR NAME = 'Grimbergen' OR NAME = 'Kro Demi' OR NAME = 'Kro Pinte'
OR NAME = 'Pelforth' OR NAME = 'Skoll' OR NAME = 'BrewDog Punk IPA' OR NAME =
'Tigre Bock' OR NAME = 'Troll Pinte' OR NAME = 'Troll Demi' OR NAME = 'Cuvée des trolls'
OR NAME = 'Paix Dieu 33cL' OR NAME = 'San Miguel' OR NAME = 'Ambrée pinte' OR NAME
= 'Ambrée demi';
```

```
DROP TABLE IF EXISTS 'Not_Beer';
```

```
CREATE TABLE 'High_Degree_Beer' AS SELECT babar_server_purchase.* FROM babar_server_purchase
JOIN babar_server_product ON babar_server_purchase.product_id = babar_server_product.id WHERE
NAME = 'Smirnoff' OR NAME = 'Pastis' OR NAME = 'Hard' OR NAME = 'Kir' OR NAME
```

= 'Cocktail Hard' OR NAME = 'Shot' OR NAME = 'Rouge Pinte' OR NAME = 'Rouge Demi'
 OR NAME = 'Sangria' OR NAME = 'Granita Hard' OR NAME = 'Hard Qualité' OR NAME =
 'JagerBomb';

2.2 Time Scale

We think that a Weekly prediction can be done, so we'll consider the sales week by week (Time series approach). If we have difficulties doing it we will consider doing it monthly but then the forecast would be much less usable. (Time series approach)

We will also maybe consider a year index. Indeed promotion aren't all the same therefor they don't consume the same quantities. Adding a bias for each promotion (which would be compute thanks to the first month of data ? september ?) could maybe increase the accuracy of our estimations.

When to start ? Beginning of 2011 or rentree 2011 ?

2.2.1 Sorting By week

For each week we will sum the sales for each categories. The weeks will be register by a index of week from 1 to 52, and a year. So 2 keys.

I don't think that I can sort the data by week by using SQL, so I'll do a python code that do that. The results will be a table with the sales for each category for each week.

Week	HighDegree	NormalDegree	SpecialBeer	NotBeer
15/03/2014 – 22/03/2014	60	82	31	15
23/03/2014 – 30/03/2014	51	74	29	25
31/03/2014 – 6/04/2014	74	90	50	38
07/04/2014 – 14/04/2014	56	77	23	20

2.3 Other used Data

In order to do our prediction we thing about using the following features :

- Day before next holiday
- Day before next exam
- Day after exam or holiday ?
- Day before next large event ?

We think that those are the main influences on whether people buy more or less drinks in a bar.

2.3.1 Holidays

2.3.2 Exams

Chapter 3

Model Selection

Now the core of our work is to choose the best method to predict from our data. Our goal is to look in the dataset for features such as trends, cyclical fluctuations, seasonality, and behavioral patterns.

Here are the algorithms that we read about and could be used to forecast sales :

- KNN Regressor
- Random Forest
- Neural network

3.1 KNN Regressor

3.2 Random Forest

3.3 Neural Network

Chapter 4

Python Code

4.1 Data Manipulation

The data is stored in Données.

4.1.1 Sales DataFrame

This file gather the function necessary to create a data frame gathering all sales information for each week. There are (by order of apparition) the functions.

- **textbfManageNullWeek** : It add rows of zeros for week with no sales.
- **textbfProductSales** : This function takes the sale of a product dataframe in argument and gives an array with the usefull categories, ie for each product it gives a array of the form : *Year|Weeknumber|Numberofsales* (And the null row are filled by ManageNull Week).
- **textbfCreateSalesArray** : opens thee CSV files of the sales and made a dataframe for each table. Then it collect the number of sales for each week by using the two previous functions. It returns a array containing the each array of each product.
- **CreateSalesData** : It take the precedent array in argument and creates an array for each year of the wanted form, ie for each week the sales of each product.
- **CreateSalesFrame** : This function is the gathering of all this file, it returns the previously described data frame by using all the functions.

4.1.2 Holidays and Exams

According to us, one of the most important parameter is the holidays. We will consider the number of weeks (or days ?) to the next holiday (and after ?). The data is gathered in a csv file with for every week if there was a holiday (indicated by a 1) or if there wasn't (indicated by a 0). Our goal is to treat this data in order to have for each week the distance to the next holiday with the convention that the holiday week has a distance of 0. For the exam we are considering the option of pondering the distance to the next exam depending on the number of exams that occurs in a week. But how to do that ?

- `openHandEData()` : This function just opens the csv file and make it a Dataframe
- `distanceToNextHoliday()` : This function creates the counter for holidays and add it to the previously made dataframe
- `distanceToNextExam()` : This function creates the counter for holidays and add it to the previously made dataframe

Chapter 5

Development Ideas

Chapter 6

Inspirations

<https://towardsdatascience.com/sales-forecasting-from-time-series-to-deep-learning-5d115514bfac> : Forecasting principles and basis

<https://medium.com/analytics-vidhya/walmart-sales-forecasting-d6bd537e4904> : Forecasting at Walmart