

Super Wine

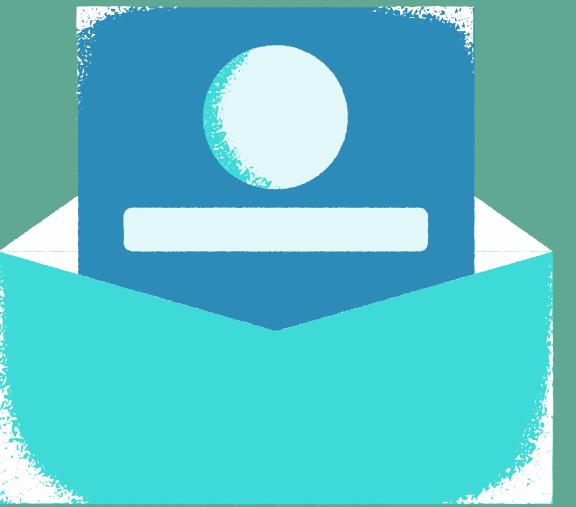
Recommender

Barbara Kharkheli

20

21

Goals of the project



Provide insights into wine
export, import, production
and consumption patterns

Part One



Develop a ML model that
suggests wines based on
meal images

Part Two



Project Steps

O1

**Data Gathering from
Different Sources**

O2

**Data
Wrangling/Cleaning**

O3

**Database Creation
and ERD**

O4

**API Creation and
Documentation**

O5

**EDA and
Visualisations**

O6

**Machine Learning
and Streamlit App**



Project Management

Trello Board

The Trello board is titled "WineProject". It features five columns:

- Planning & Data Collection**:
 - Define Project Objectives and Goals
 - Collect Data from OIV
 - Collect Data from INSEE
 - Collect data from Data.Gouv.Fr
 - Collect Data from AgRest
 - Web-scraping
 - Register for Wine-Searcher API
 - Get data from OpenFoodFacts API
 - + Add a card
- Data Cleaning & EDA**:
 - Choose the list of questions to answer
 - Remove null columns and duplicates
 - Reshape tables, analyse columns, drop columns etc.
 - Define questions to answer
 - Visualize in Tableau
 - + Add a card
- Database Creation**:
 - Normalize data
 - MySQL ERD
 - SQL Queries
 - BigQuery
 - Create an API
 - + Add a card
- Machine Learning/Wine Recommender**:
 - Check how to use GPU's
 - Set up a Streamlit Website
 - Download Food 101 image collection
 - Train Food 101 model
 - + Add a card
- Report & Pres**:
 - Choose a Presentation template
 - Write a report for Tuesday
 - Build up a pres on Canva
 - Rehearse
 - + Add a card

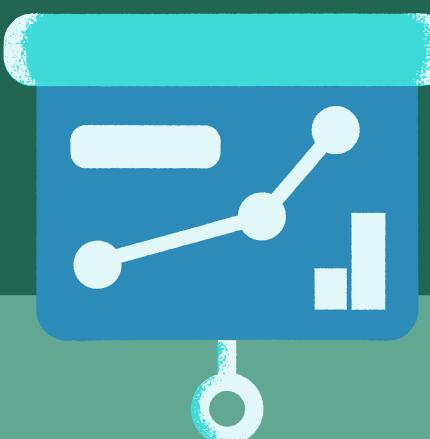
At the top right, there are buttons for "Power-Ups", "Automation", "Filters", and "Share".

01

Data Gathering

To achieve the project goals, I will collect data from multiple sources, ensuring a comprehensive and robust dataset for analysis and model training





Flat File Sources:

OIV - Organisation Internationale de la Vigne et du Vin

Data.Gouv.Fr

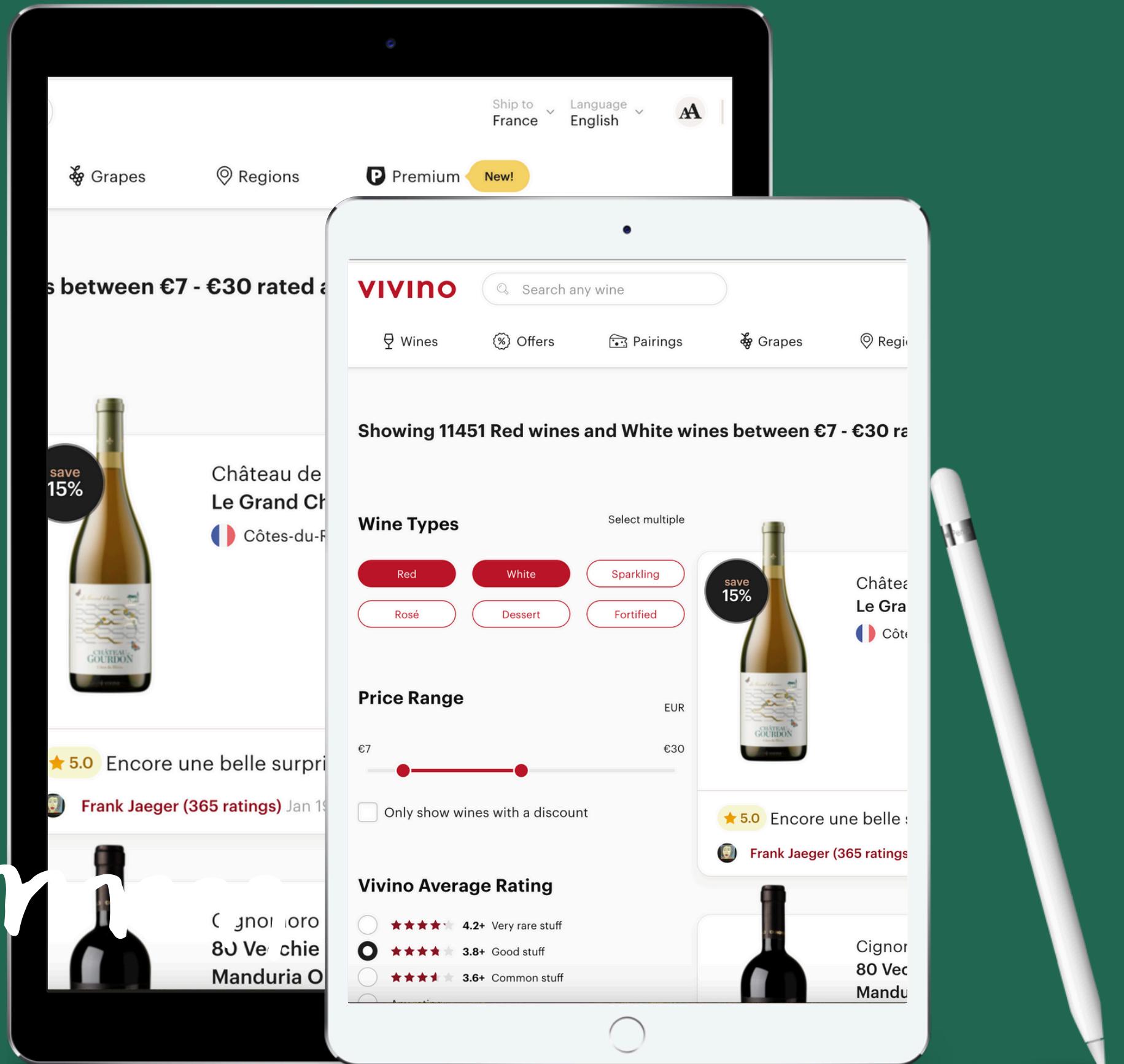
AgReste (Ministry of Agriculture)

INSEE

The screenshot displays the OIV's data visualization interface. At the top, there are navigation links in French: Qui nous sommes, Ce que nous faisons, Ce que nous proposons, Médias, and Contact. Below the header is a large banner with the text "Base de données" and "Ce que nous faisons". The main content area includes a map of Asia and a table with the following data:

Continent	Region/Country	Product	Variable
Asia	Afghanistan	Dried Grapes	Consumption
Asia	Afghanistan	Table Grapes	Consumption
Asia	Afghanistan	Dried Grapes	Exports
Asia	Afghanistan	Fresh Grapes	Exports
Asia	Afghanistan	Fresh Grapes	Imports
Asia	Afghanistan	Dried Grapes	Production
Asia	Afghanistan	Fresh Grapes	Production
Asia	Afghanistan	Table Grapes	Production
Asia	Afghanistan	Dried Grapes	Consumption
Asia	Afghanistan	Table Grapes	Consumption





The graphic features a teal header with a white line graph icon. Below it is a green rounded rectangle containing the text 'Vivino.com - Webscraping'. At the bottom, there are two sections: 'Top 100 wines worldwide' and 'Retrieve Name, Producer, Country, Price'.



Data from Open Food Facts API

Fetch up to 5000 wine records in France

Create a dictionary with basic information
(Name, Category, Ecoscore, Country)

```
canva.com
world.openfoodfacts.net/cgi/search.pl?action=process&json=1&fields=product_name,brands,categories&page_size=100&page=1&query=Impression%20%C3%A9l%C3%A9gante

[
  {
    "count": 7286,
    "page": 1,
    "page_count": 100,
    "page_size": 100,
    "products": [
      {
        "brands": "Saguaro",
        "categories": "Boissons, Boissons alcoolisées, Vins, Boissons gazeuses, Eaux, Eaux de sources, Gazeuses, Vins pétillant, en:Boissons, Boissons gazeuses, Eaux, Eaux de sources, Eaux minérales",
        "product_name": "Finement pétillante"
      },
      {
        "brands": "Ovomaltine",
        "categories": "Boissons, Boissons alcoolisées, Snacks, Snacks sucrés, Vins, Cacao et dérivés, Chocolatées, Alsace Grand Cru, Alsace Grand Cru Saering",
        "product_name": "Ovomaltine - Vitamin & Mineral Bars, 5 ct 100g (3.5oz)"
      },
      {
        "brands": "Danone",
        "categories": "Boissons, Boissons alcoolisées, Vins, Eaux, Vins français, Eaux de sources, Fruits et légumes",
        "product_name": "Ain Saiss Eau Minerale Naturelle"
      },
      {
        "brands": "Saveurs de nos régions",
        "categories": "Plant-based foods and beverages, Beverages, Plant-based foods, Alcoholic beverages, Fruits based foods, Plant-based spreads, Sweet spreads, Fruit and vegetable preserves, Wines and beers",
        "product_name": "Fruchtaufsteich Klementine-Orange"
      }
    ]
  }
]
```

02

Data Cleaning

The most time-consuming and crucial part was managing data quality. This included handling null values and duplicates, adjusting data types, removing unnecessary columns, reshaping tables, and ultimately saving the cleaned tables to CSV files.

Exemple of cleaning and wrangling OIV table

Data Source: OIV - Organisation Internationale de la Vigne et

- Display the shape of the DataFrame.
- Display the number of missing values, duplicate rows, and data types of each column.
- Remove unnecessary columns
- Display the count of unique values in each column to check for abnormalities
- Exclude the years 2022 and 2023 from rows (due to lack of information)

The screenshot shows a Jupyter Notebook environment. On the left is a file tree with notebooks like EDA_Wine.ipynb, MachineLearning, and training_model.ipynb. The main area has two data frames. The top data frame has 116 rows and 5 columns, showing data for Europe: France, Production, Consumption, Exports, Imports, and Year. The bottom data frame has 113 rows and 5 columns, showing data for Global: Global, Consumption, Exports, Production, and Year. Below the data frames are three code cells:

```
consowine_global = consowine[consowine['Continent'] == 'Global']
consowine_global
```

```
...    Continent Region/Country Variable Year Quantity
43507   Global      Global Consumption 1995.0 227425.0
43508   Global      Global Exports 1995.0 55016.0
43509   Global      Global Imports 1995.0 51771.0
43510   Global      Global Production 1995.0 250874.0
43511   Global      Global Consumption 1996.0 221646.0
...
43615   Global      Global Consumption 2022.0 231600.0
43616   Global      Global Exports 2022.0 107515.0
43617   Global      Global Production 2022.0 262107.0
43618   Global      Global Consumption 2023.0 221219.0
43619   Global      Global Production 2023.0 237340.0
```

```
consowine_global = consowine_global.drop(columns = ['Region/Country', 'Continent'])
consowine_global = consowine_global[consowine_global ['Year'] != 2022]
consowine_global = consowine_global[consowine_global ['Year'] != 2023]
```

Python Code

03

Database Creation and ERD

The 'wine_schema' database was created in MySQL Workbench to store 11 tables related to the wine market analysis which I collected using flat files, API and web scraping

Wine Schema



Ironhack
wine_schema

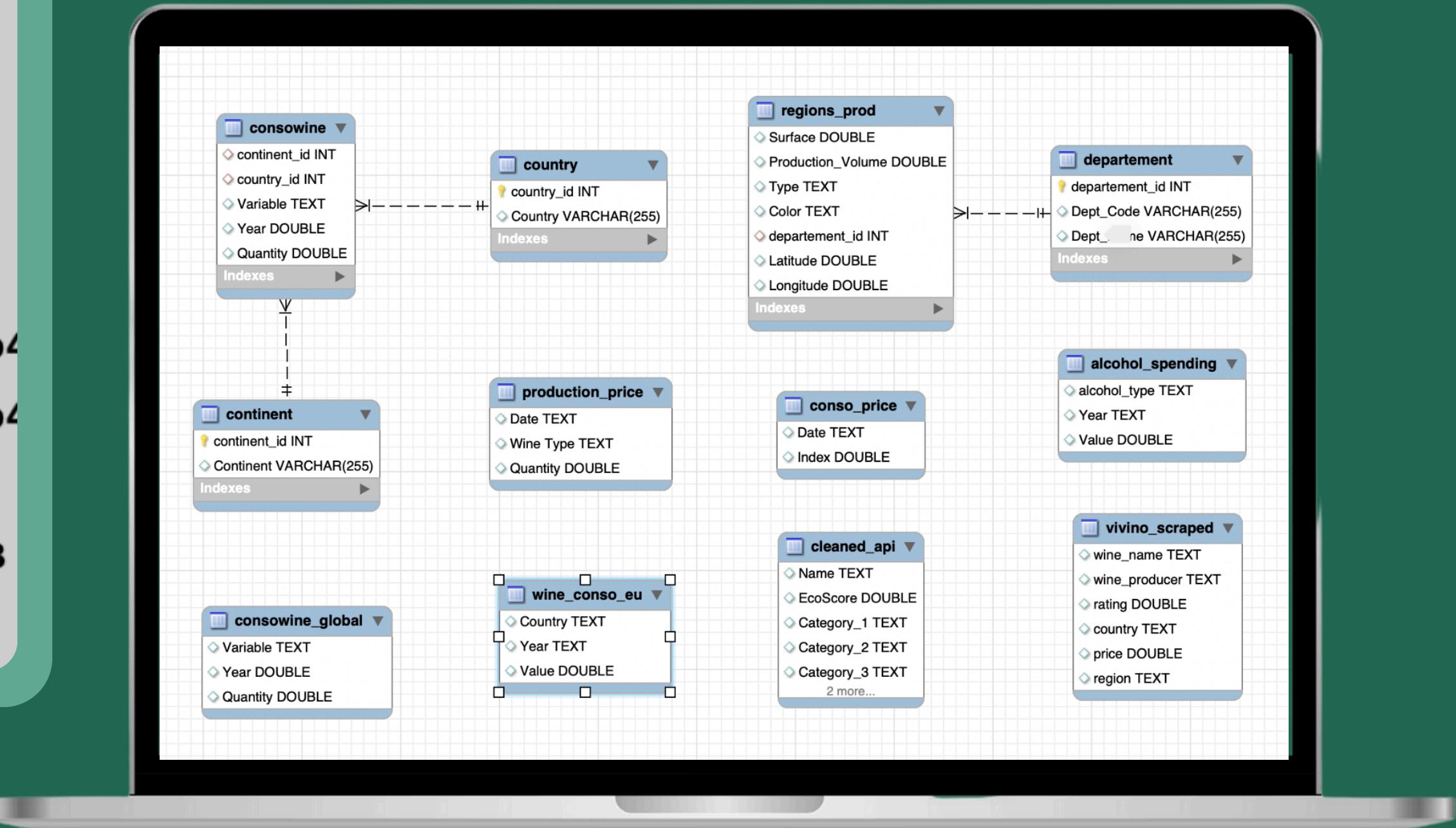
Schema Details

Default collation:
utf8mb4

Default characterset:
utf8mb4

Table count:
11

Database size (rough estimate): **3.8 MiB**



MySQL ERD

04

API Creation and Documentation

I created a Restful API to display the data.

API Creation

Endpoints of my API:

wine_consumption

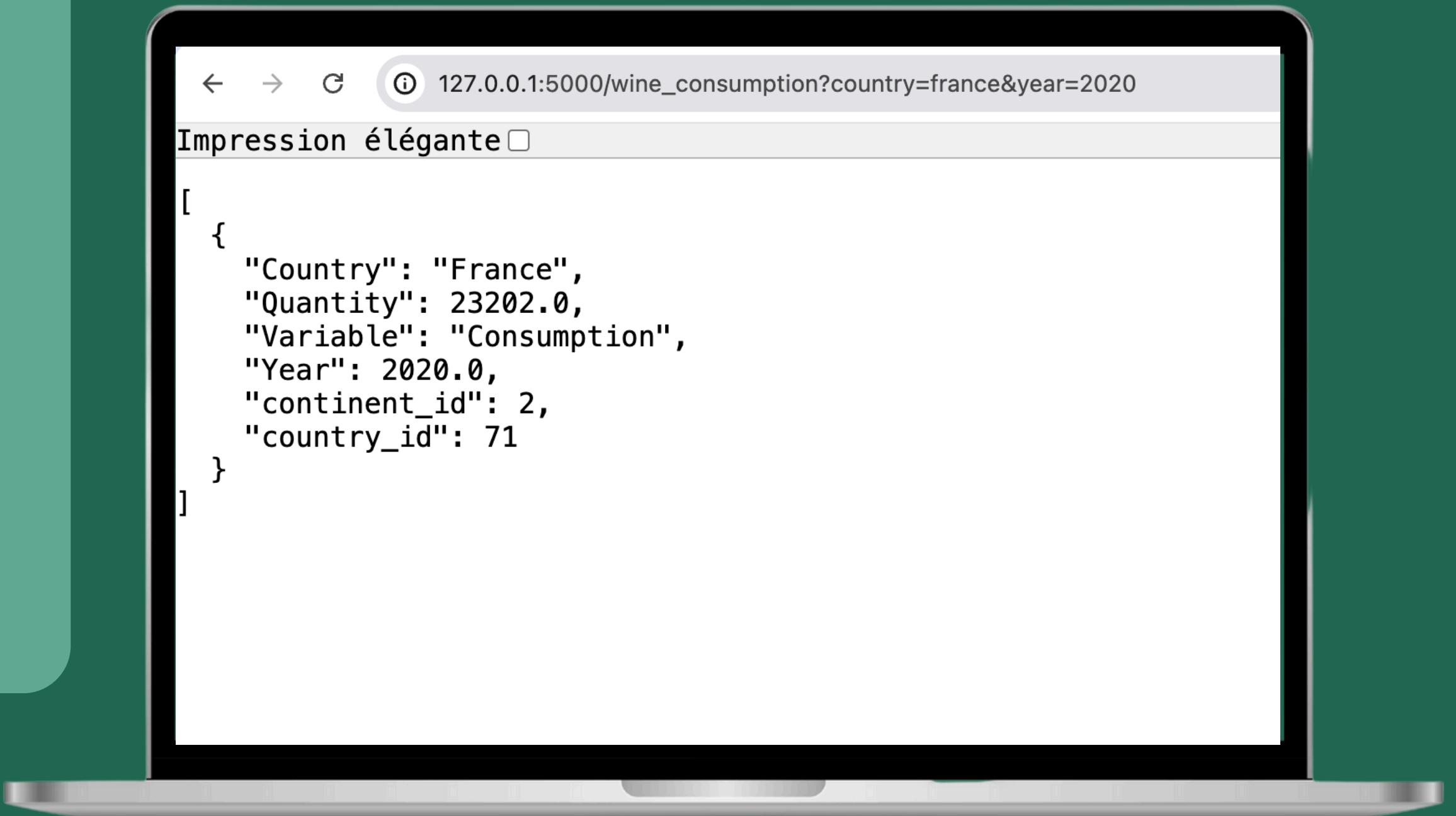
parameters: country & year

wine_production

parameters: country & year

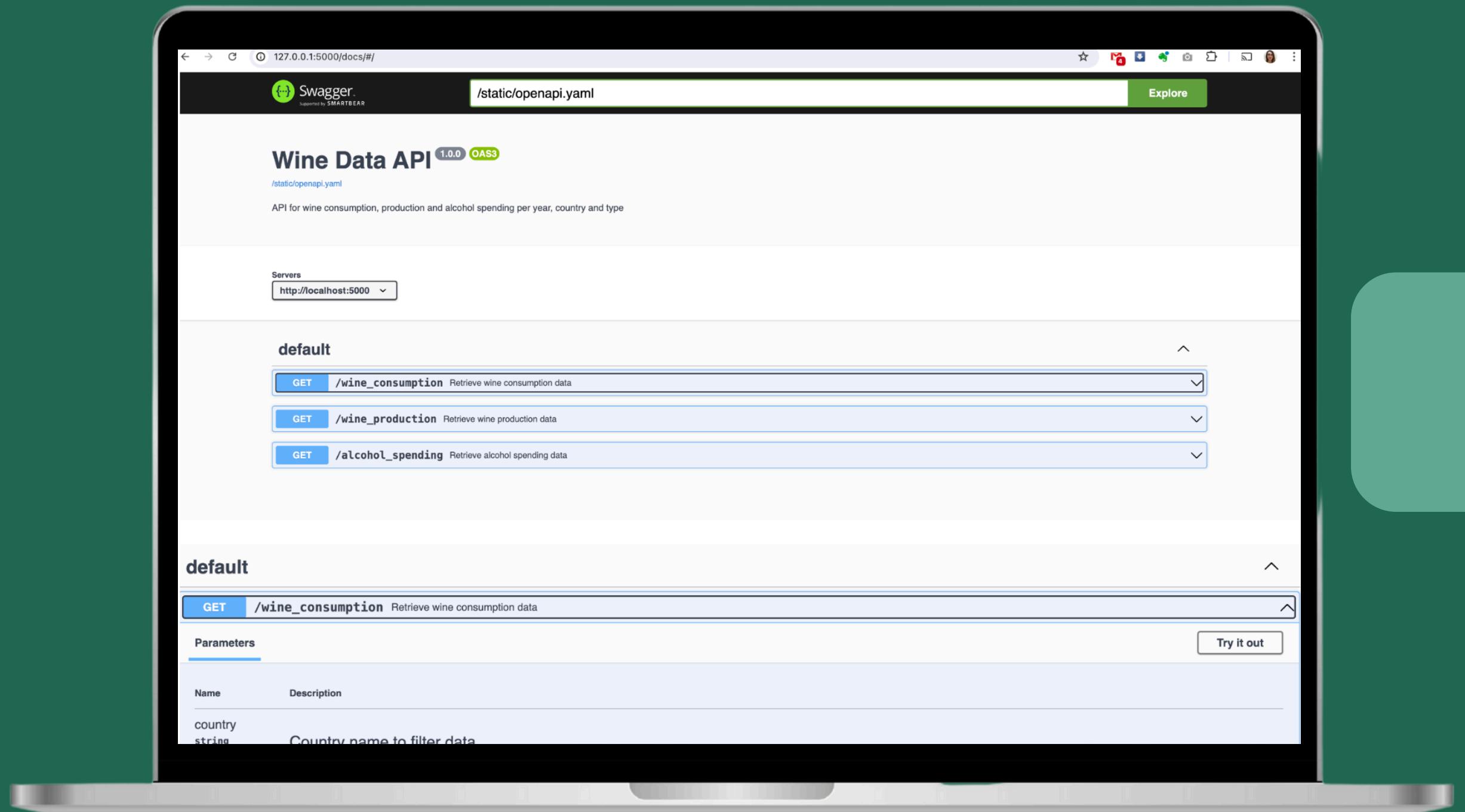
alcohol_spending

parameters: type & year



API Demonstration

API Documentation



I used Swagger Open API
to create a YAML file

SWAGGER Docs

05

EDA and Visualisations

In this section I will present you with some interesting data exploration and visualisations done in Tableau.

Wine in France

796 K

Hectares of
vineyards
(Second to Spain)

4.69 B

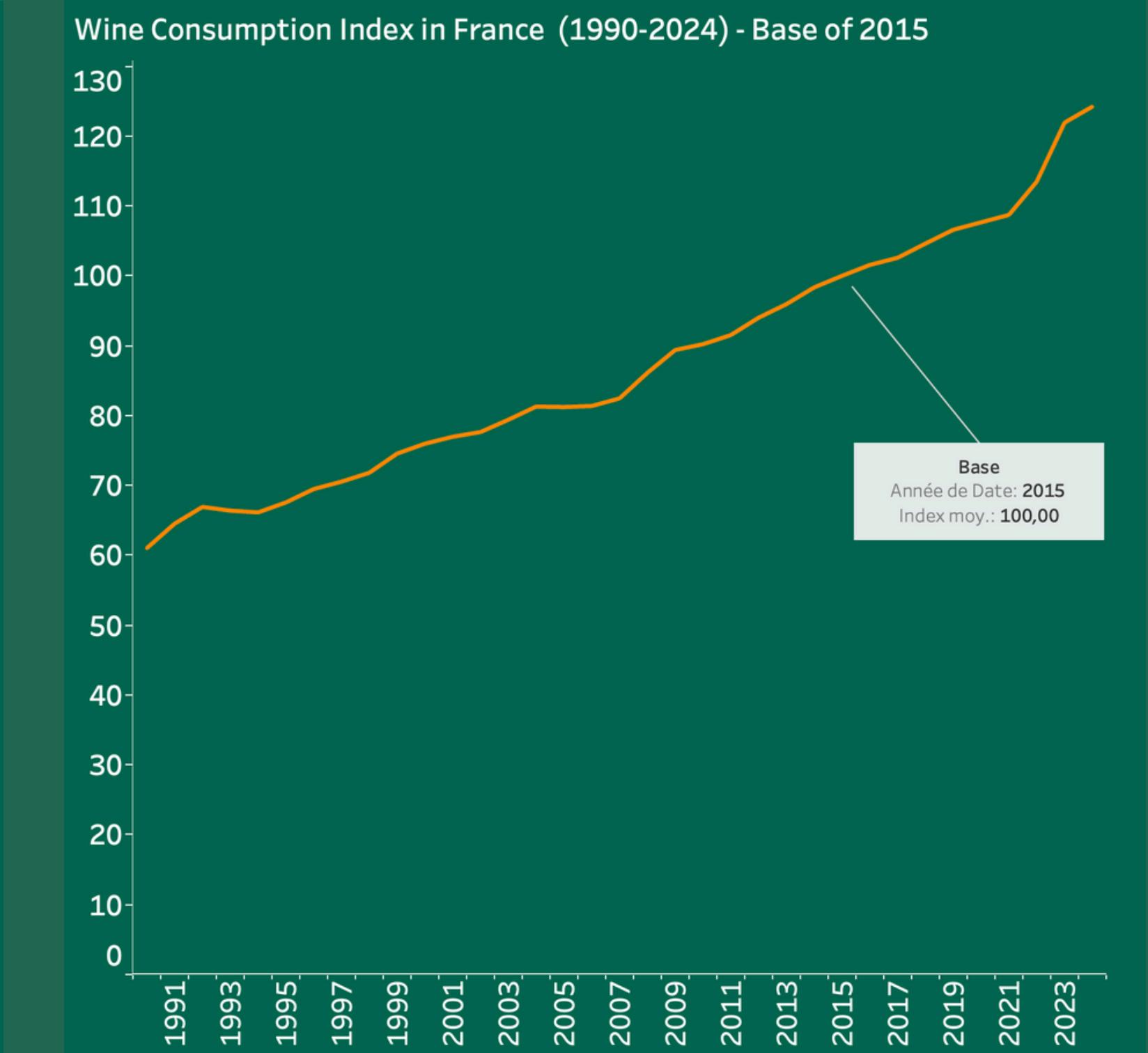
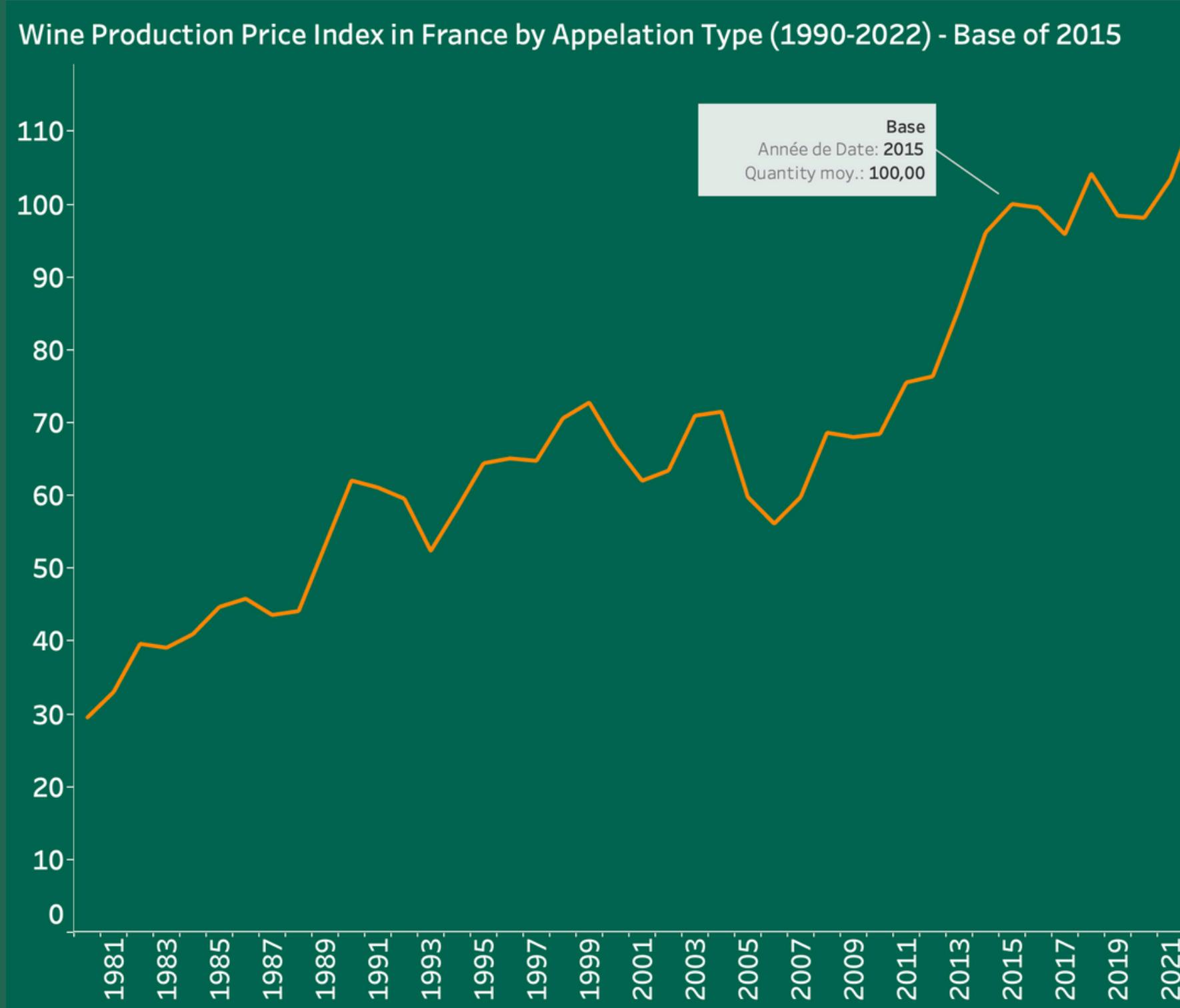
Liters of wine
produced
(Second to Italy)

1.3 B

Liters of wine
exported
(Third to Italy and
Spain)

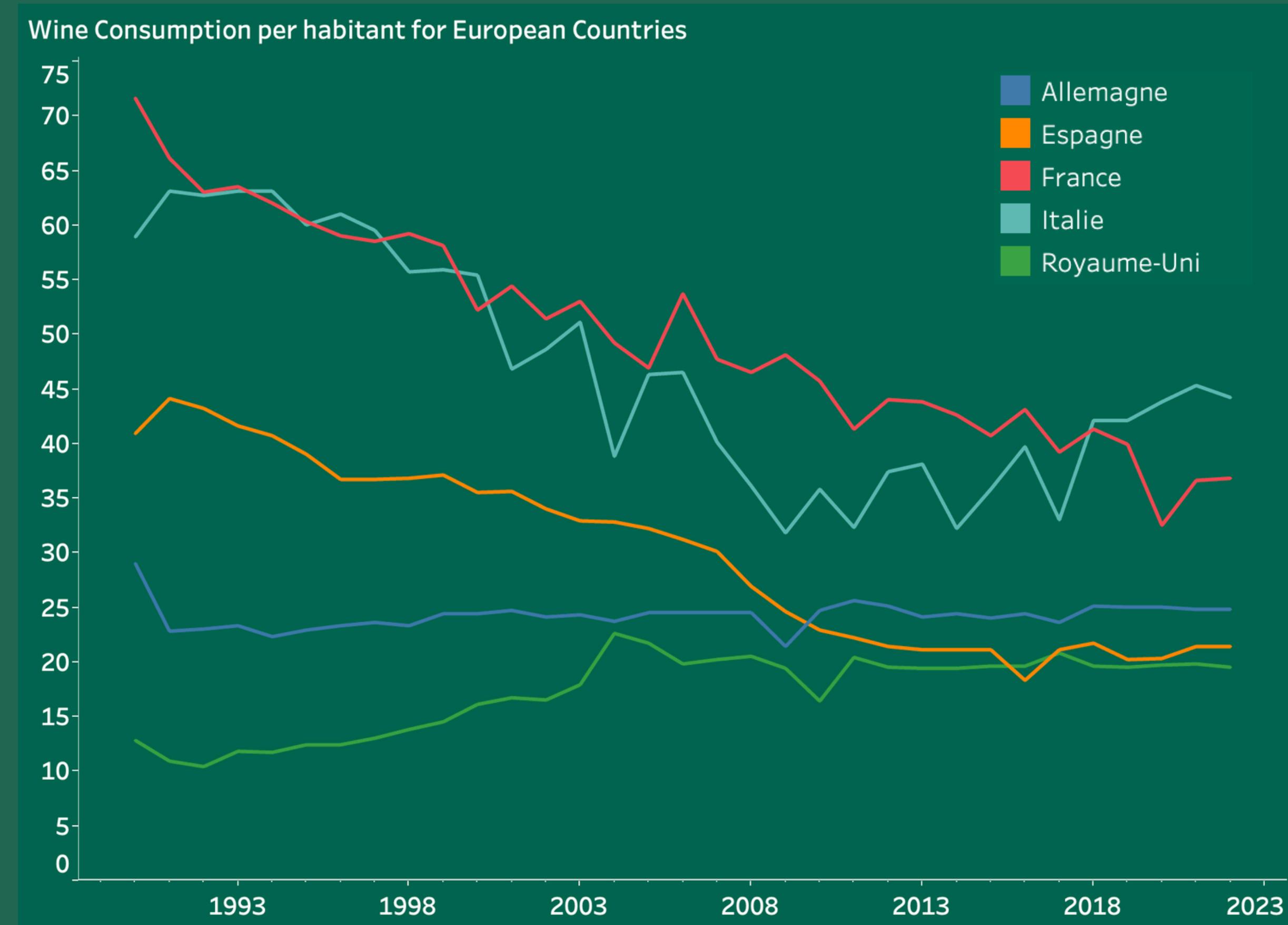
Price Trends in France

We see clear increase in production and consumption prices

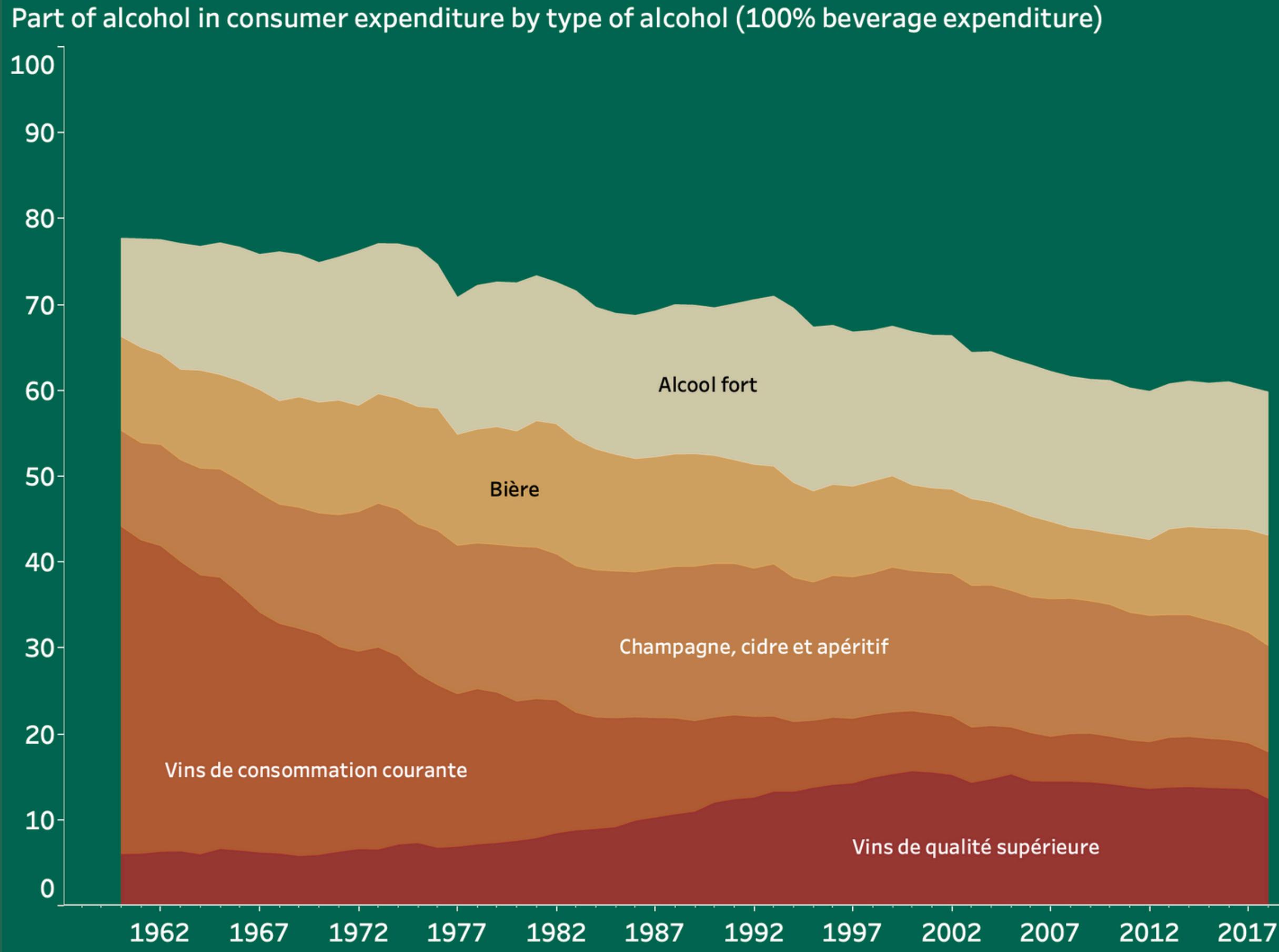


Consumption per habitant

We observe that consumption in France is clearly decreasing compared to other countries.



Alcohol Spending



We can observe that customers shifted from consuming table wine to consuming higher quality wine.

06

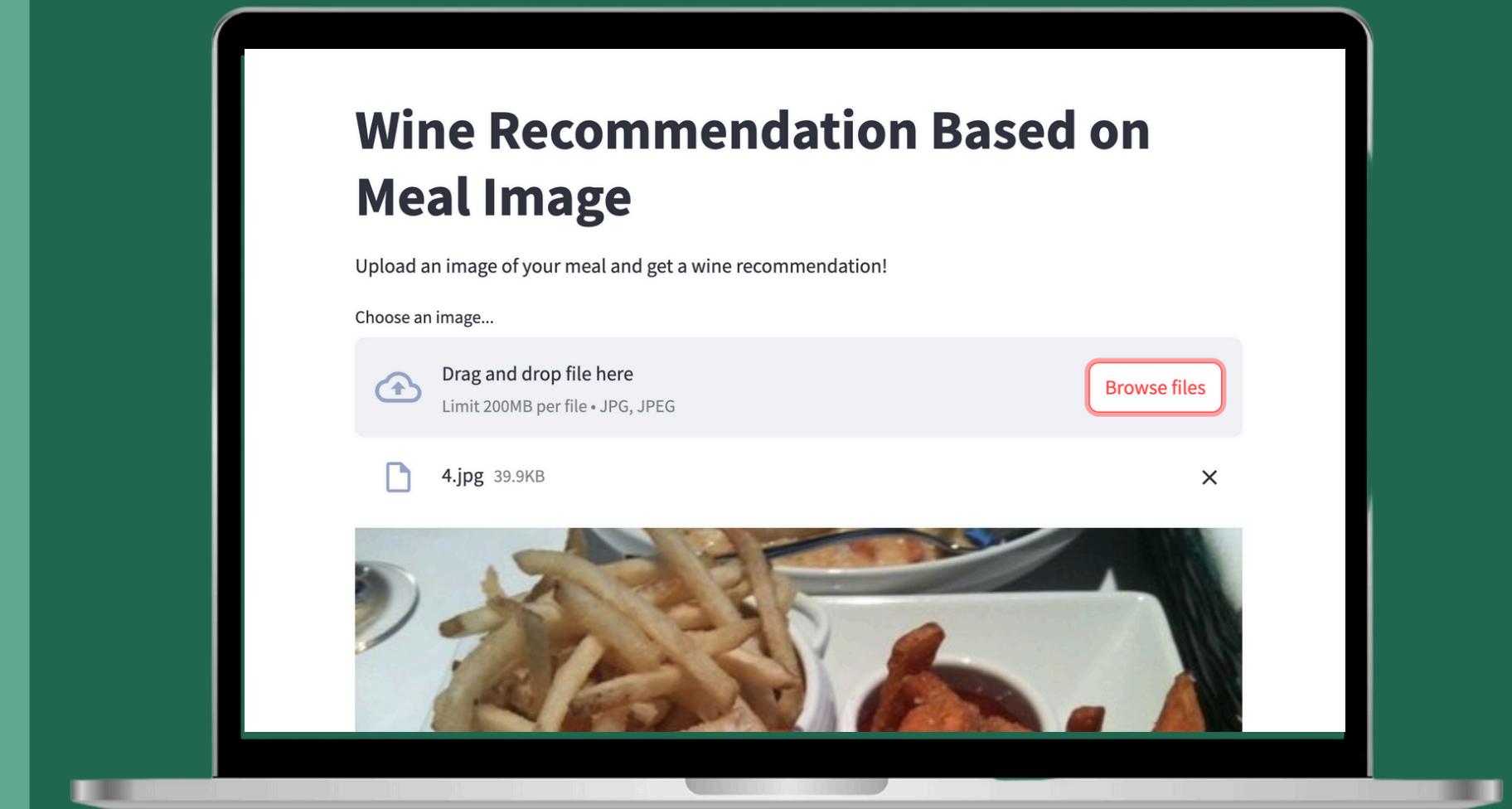
Machine Learning

This project aims to develop a Streamlit application that integrates a food recognition system with a wine recommendation engine.

Overview of ML

This project aims to develop a Streamlit application that provides users with personalised wine recommendations based on images of their meals by leveraging machine learning and deep learning techniques.

The project uses the FOOD 11 image database to train a convolutional neural network (CNN) model for accurate food recognition.



First attempt on the app

Food 101 Fail



pynb

```
#Adding custom Layers
x = model.output
x = Flatten()(x)
x = Dense(101*2, activation="relu")(x)
x = Dense(101*2, activation="relu")(x)
predictions = Dense(101, activation="softmax")(x)
model_final = Model(input=model.input, output=predictions)
model_final.compile(loss="categorical_crossentropy", optimizer="adam")
model_final.load_weights(models_filename)
```

⌚ 754m 59.7s

...
2024-07-08 19:03:03.954266: I metal_plugin/si
2024-07-08 19:03:03.954313: I metal_plugin/si
2024-07-08 19:03:03.954325: I metal_plugin/si
2024-07-08 19:03:03.954599: I tensorflow/core

101000 images in 101 classes...
13 hours and still running...

Tuning the parameters to reduce time

Food 11 Dataset

16 643 food images grouped in 11 major food categories:

Bread, Dairy product, Dessert, Egg, Fried food, Meat, Noodles-Pasta, Rice, Seafood, Soup, Vegetable-Fruit

I tried two already trained lightweight convolutional neural network (CNN)

MobileNetV2

VGG16

30 epochs

Python Code

```
# Use a smaller model for faster training
base_model = MobileNetV2(input_shape=(224, 224, 3), weights='imagenet', include_top=False)
num_classes = len(labels) # Ensure num_classes matches the number of classes in the data

x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(128, activation='relu')(x)
x = Dropout(0.5)(x) # Adding dropout for regularization
out = Dense(num_classes, activation='softmax')(x)

model = Model(inputs=base_model.input, outputs=out)

# Freeze base model layers
base_model.trainable = False

# Compile model
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

# Train model
num_epochs = 10 # Increased number of epochs for better accuracy
STEP_SIZE_TRAIN = min(len(train_generator), train_generator.n // train_generator.batch_size)
STEP_SIZE_VALID = min(len(valid_generator), valid_generator.n // valid_generator.batch_size)
STEP_SIZE_TEST = min(len(test_generator), test_generator.n // test_generator.batch_size)

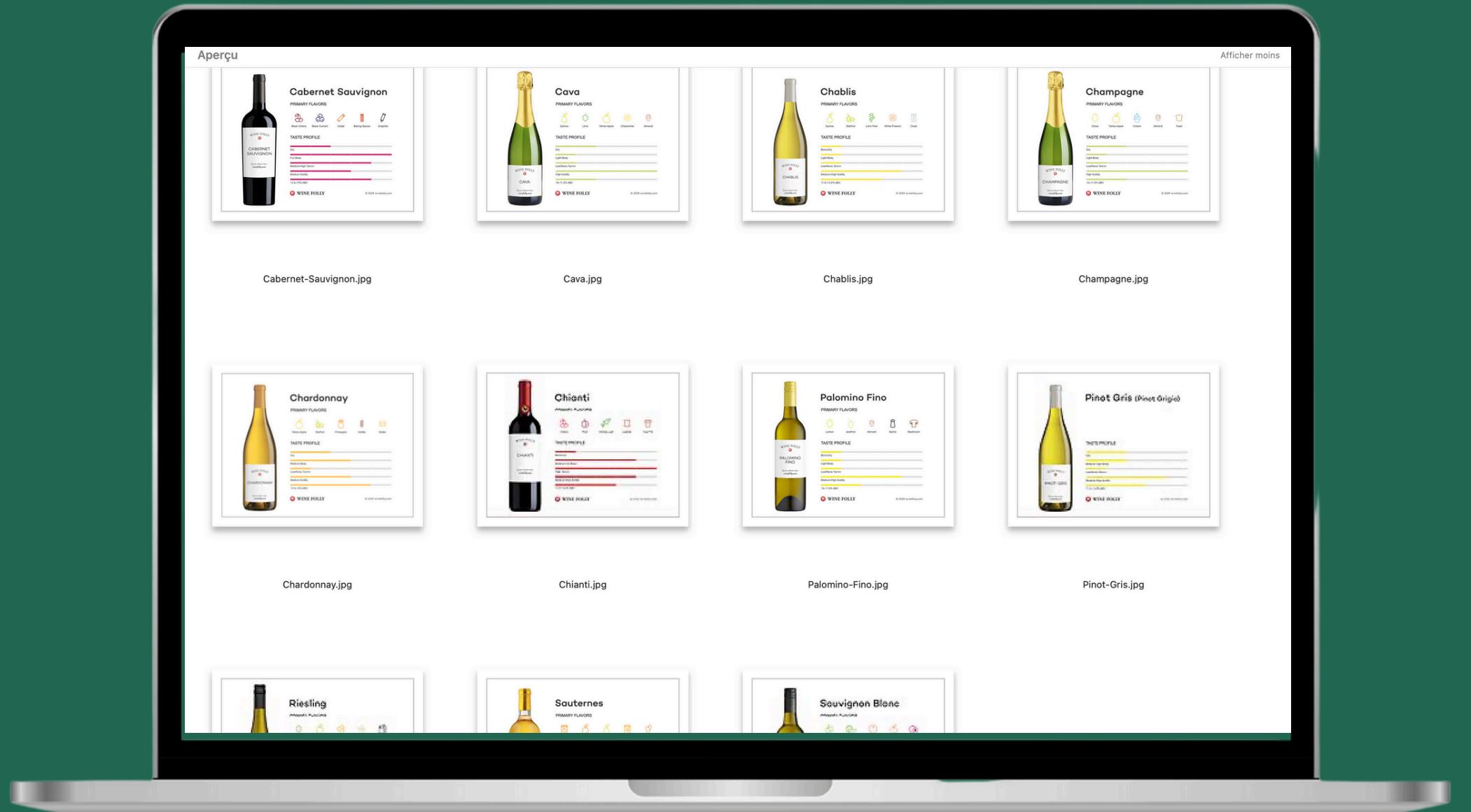
history = model.fit(
    train_generator,
```

0,77

Accuracy Score

(and possibility to do better with more time and processing power)

Creating a simple recommendation dataframe



Wine Cards

Dataframe contains:
Food Class
Wine recommendation
Wine Card with more information

I linked it to the
recognition model for
my Streamlit
application

Demo

The image shows a tablet device with a black frame and a silver stand, displaying a mobile application. The app has a teal and orange color scheme. At the top, there's a header with the text "Super Wine" and "Recommender". Below this, a sub-header says "Based on your food". The main area features several orange cards with icons: one with a star and grapes, another with a dollar sign, one with the number "20", one with "21", and one with a person icon. At the bottom, there's a file upload section with a placeholder "Choose an image...", a "Drag and drop file here" button with a cloud icon, a "Limit 200MB per file • JPG, JPEG, PNG" note, a "Browse files" button, and a preview thumbnail of a pasta dish.

Super Wine Recommender

Upload an image of your meal and get a wine recommendation!

Choose an image...

Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG

Browse files

67700_RichPastaforthePoorKitchen_ddmfs_4x3_2284-220302ec8328442096... 141.6KB X

Super Wine Recommender



Challenges

These are the areas where I faced the most difficulties.

O1

**Finding relevant data
on internet. Cleaning
and making it usable.**

O2

**Web-Scraping with
Selenium due to high
protections of the
page**

O3

**Understanding the
food recognition
models and making it
work**



Next Steps

O1 Train the model on more images to increase accuracy

O2 Develop a more sophisticated database for wine recommendation

O3 Improve the application UX design

O4 Commercialize it :)

Thank You

