IRON HACK

Data Analytics

Wine Market Analysis

Barbara KHARKHELI

July, 2024

Introduction

The wine industry in France is a cornerstone of the country's cultural and economic landscape. Known for its diverse wine regions and high-quality production, France remains a global leader in wine exportation, consumption, and production. With its rich history and tradition, French wine holds a special place in the hearts of wine enthusiasts around the world.

In this project, I aim to develop a comprehensive analysis of the wine market in France. By leveraging various data collection methods, including web scraping, API integration, and database management, I seek to provide detailed insights into wine consumption, export, import, production and price tendencies.

Goal

The goal of this project is to:

- Provide insights into wine export, import, production and consumption patterns
- Try to develop a recommendation system that suggests wines based on meal images

Objective

The objective is to collect, clean, and analyze data related to wine production, consumption, and prices in France. By doing so, I will develop a robust database that provides valuable insights into the wine industry. The final deliverable will include detailed reports and visualizations that highlight key trends and patterns in the wine market and a web application for wine recommendations.

Use Case

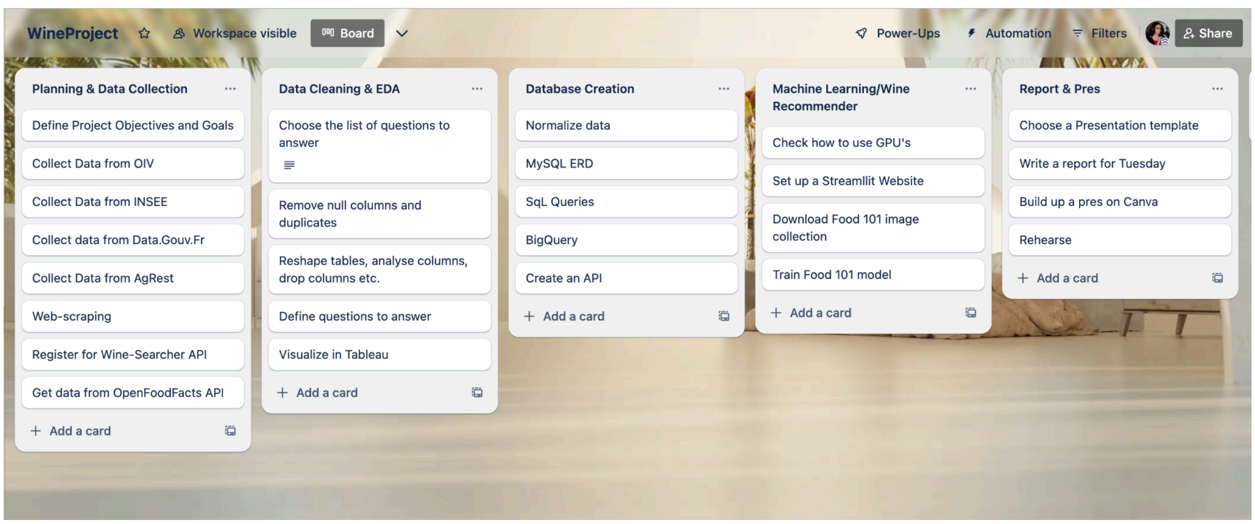
The primary use case for this project is to develop a system that can analyze and recommend wines to users based on their meal choices. The recommendation system will utilize image recognition to identify meals and suggest the best matching wines, thereby enriching the dining experience. This system will be particularly useful for wine enthusiasts, restaurants, and retailers who want to offer personalized wine suggestions to their customers.

Project Management

Effective project management is crucial for the successful completion of this project. I will use Trello to track and manage project tasks and milestones. Trello will help me organize data collection, cleaning, analysis, database creation, and API development tasks, ensuring timely delivery and that no important details are overlooked.

By keeping track of all activities and deadlines, Trello will ensure that each phase of the project is completed efficiently and on schedule. Regular updates and reviews will be conducted to monitor progress and address any challenges promptly.

My Trello Board



Data and data sources

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

- Flat Files: I will import data from CSV and Excel files, including detailed consumption and production statistics. These files will provide a structured foundation for understanding trends and patterns in the wine industry.
- API Integration: By utilizing APIs like the Open Food Facts API, I can obtain real-time, structured data on various wine products. This integration will enrich the dataset with up-to-date information on wine types, eco scores, and other relevant attributes.
- Web Scraping: I will extract data from Vivino , the biggest wine portal in the world to check how many French wines figure in the top wines worldwide
- Database Management: The collected data will be stored and managed in a relational database. This approach will facilitate efficient querying and analysis, allowing for the seamless integration of different data sources and the extraction of meaningful insights.
- Big Data Systems: To handle large and complex datasets, I will denormalize and store data in a big data system. This will enable scalable data processing and advanced analytics, ensuring that the system can handle high volumes of data and provide timely insights.

By leveraging these diverse data sources, I will create a comprehensive and dynamic dataset that supports detailed analysis and robust machine learning models.

Data Sources

OIV - Organisation Internationale de la Vigne et du Vin

Entity Description: The OIV is an international organization that focuses on scientific and technical aspects of viticulture and winemaking. It provides a global forum for the exchange of information and standards related to vine cultivation, wine production, and the wine market. The OIV collects, analyzes, and disseminates data on wine production, export, import, and consumption across various countries, making it a crucial resource for understanding global wine industry trends.

URL: OIV Data

Information::

Production: Contains data on the volume of wine produced by different countries.

Export: Records the quantity of wine exported by various countries.

Import: Details the amount of wine imported by different countries.

Consumption: Provides statistics on wine consumption per country and type.

Data.Gouv.Fr

Entity Description: Data.Gouv.Fr is the French government's open data platform. It provides access to a wide range of public datasets, including those related to agriculture and viticulture. This platform offers annual statistics on viticulture, including data on wine stocks and harvests by region in France. Additionally, it provides production quantities by regions of France and by appellation type. This data is essential for analyzing regional production trends and understanding the dynamics of the French wine industry.

URL: Statistiques viti-vinicoles

Information:

Annual Harvests: Records of wine harvest quantities by region and appellation type.

Production Quantities: Detailed production data by region and appellation.

AgReste

Entity Description: AgReste is the statistical department of the French Ministry of Agriculture. It provides detailed statistical data on various aspects of agriculture, including food consumption, production, and prices. The datasets from AgReste include:

- **Consumption by Habitants:** Data on wine consumption by inhabitants, including the ranking of European countries by wine consumption (liters per inhabitant).
- **Prices at Production:** Data on wine production prices, with a base year of 2015.
- **Consumer Price Indices (CPI):** Monthly and annual consumer price indices for France, with a base year of 2015.

This data is valuable for understanding domestic consumption patterns and the economic factors influencing the wine market in France.

URLs:

Consumption by Habitants

Prices at Production

Consumer Price Indices

INSEE

Entity Description: INSEE (Institut National de la Statistique et des Études Économiques) is the national statistics bureau of France. It provides comprehensive statistical data on the French economy and society. The dataset on household expenditures on beverages, including wine, offers historical insights into how spending patterns on alcoholic beverages have evolved over time. This information is useful for analyzing the economic impact of wine on French households and comparing it with other beverages.

URL: INSEE Household Expenditure

Information:

- **Household Expenditure:** Data on the spending of families on different types of beverages, including wine, beer, and spirits, over the last 60 years. (Les dépenses des ménages en boissons depuis 1960)

Open Food Facts API

Entity Description: Open Food Facts is a collaborative project that collects and shares data on food products from around the world. The Open Food Facts API provides access to a database of food products, including detailed information on ingredients, nutritional values, and other product attributes. For this project, the API is particularly useful for obtaining structured data on various wine products, which can be used to enhance the wine recommendation system.

URL: Open Food Facts API

Information:

Product Information: Contains detailed data on wine products, including names, brands, categories, ingredients, and eco scores.

Vivino.com - Webscraping

Entity Description: Vivino is a popular wine review and rating website that provides comprehensive information on wines from around the world. By web scraping Vivino, I will gather data on the top 100 wines worldwide to see how many French wines are included. This data will help in analyzing the global standing of French wines in terms of popularity and ratings.

URL: Vivino Top 100 Wines

Information:

Top 100 Wines: Data on the top 100 wines worldwide, including details such as wine name, producer, rating, country, and price.

By leveraging these well-established data sources and their respective datasets, I aim to gather a comprehensive and diverse dataset that will support detailed analysis and the development of a robust wine recommendation system.

Data collection and Cleaning

Wine Statistics Dataset

Data Source: **OIV - Organisation Internationale de la Vigne et du Vin**

- Load an Excel file containing wine consumption data into a DataFrame.
- Filter the data to include only rows where the product is 'Wine'
- Display the shape of the DataFrame.
- Check for missing values, duplicates, and data types
- Display the number of missing values, duplicate rows, and data types of each column.
- Display value counts for specific columns
- Display the count of unique values in the 'Continent', 'Region/Country', 'Variable', 'Year', and 'Unit' columns.
- Remove the 'Product' and 'Unit' columns from the DataFrame.
- Filter the DataFrame to include only rows where the 'Continent' is 'Global'.
- Remove the 'Region/Country' and 'Continent' columns.
- Exclude the years 2022 and 2023 (due to lack of information)
- Reset the index and remove the old index column.
- Save the global wine consumption data to a CSV file **- consowine_global.csv**

- Filter data to exclude global statistics
- Filter the DataFrame to exclude rows where the 'Continent' is 'Global'.
- Save the filtered wine consumption data to a CSV file - **consowine.csv**

Regional Production Dataset

Data Source: **Data.Gouv.Fr**

- Load a CSV file containing wine production data by department.
- Remove specific columns and rows where the department is 'TOTAUX'.
- Rename columns to more meaningful names.
- Replace missing values with zeros.
- Check for duplicates and drop rows with all zero values

- Display the number of duplicate rows and remove rows where all specified columns have zero values.
- Remove commas from string columns and convert them to float.
- Melt the DataFrame for AOP, IGP, and VSIG production volumes by color
- Reshape the DataFrame to have separate rows for each wine type and color combination.
- Rename the surface columns and add the ‘Type’ column
- Rename columns and add a new column to indicate the wine type.
- Concatenate the melted DataFrames
- Combine the melted DataFrames for AOP, IGP, and VSIG into a single DataFrame.
- Split the ‘Type_Color’ column into separate ‘Type’ and ‘Color’ columns
- Split the combined ‘Type_Color’ column into two separate columns for type and color.
- Remove the combined ‘Type_Color’ column and arrange the remaining columns in the desired order.
- Ensure the ‘Departement’ column is of string type and split it into ‘Dept_Code’ and ‘Dept_Name’.
- Merge the data with department details including latitude and longitude.
- Save the DataFrame to a CSV file **- regions_prod.csv**

Alcohol Spending Dataset

Data Source: **INSEE**

- Load an Excel file containing alcohol spending data.
- Melt the DataFrame
- Reshape the DataFrame to have separate rows for each year and type of alcohol.
- Rename columns to more meaningful names.
- Convert ‘Year’ column to datetime
- Save the DataFrame to a CSV file **- alcohol_spending.csv**

European Wine Consumption by habitant Dataset

Data Source: AgReste

- Create a DataFrame containing wine consumption data per capita for various countries over several years.
- Melt the DataFrame
- Reshape the DataFrame to have separate rows for each year and country.
- Convert the ‘Year’ column to datetime format.
- Save the processed data to a CSV file. - **wine_conso_eu.csv**

Production Price Trend Dataset

Data Source: AgReste

- Create a DataFrame containing wine production prices indexed to the year 2015.
- Transpose the DataFrame and set the first row as column headers
- Reset the index to make the index a column and convert it to datetime format.
- Rename the index column to ‘Date’.
- Melt the DataFrame
- Reshape the DataFrame to have separate rows for each wine type and date.
- Ensure the ‘Quantity’ column is in float format.
- Save the processed data to a CSV file - **production_price.csv**

Consumption Price Trend Dataset

Data Source: AgReste

- Load a CSV file containing the consumer price index for wine and fermented beverages.
- Transpose the DataFrame and set the first row as the column headers.
- Reset the index to make the index a column and convert it to datetime format.
- Rename the index column to ‘Date’.
- Convert ‘Index’ column to float
- Save the processed data to a CSV file. - **conso_price.csv**

API

Source: OpenFoodFacts API

API Data Gathering

- Define the Base URL and Parameters for Open Food Facts API
- Develop a function to fetch wine data using pagination and filters.
- Set a target to fetch up to 5000 wine records.
- Handle pagination by incrementing the page number in each iteration.
- Filter the fetched data to ensure only wine products are included.
- Use the requests library to make API calls.
- Check the response status to ensure successful data retrieval.
- Parse the JSON response to extract product information.
- Continue fetching data until the target number of wine records is reached or there are no more pages left.
- Extract relevant information from each wine product and store it in a structured format.
- Clean and format the category information for each wine.
- Create a pandas DataFrame from the list of dictionaries containing the wine data.
- Save the DataFrame to a CSV file for further processing and analysis.

API Data Cleaning

- Load the CSV file containing wine data into a pandas DataFrame.
- Remove columns that are not needed for the analysis.
- Filter rows to include only those related to ‘Boissons’, ‘Boissons alcoolisées’, and ‘Vins’.
- Filter Out Specific Categories
- Identify and remove rows where certain categories (e.g., non-wine products) are present.
- Check for Missing Values and Duplicates
- Display the number of missing values and duplicate rows in the DataFrame.
- Handle missing values appropriately
- Drop duplicate rows from the DataFrame, keeping only the first occurrence.
- Save the cleaned DataFrame to a new CSV file **- cleaned_api.csv**

Web-Scraping Vivino.Com

Data Gathering

- Imported pandas for data manipulation.
- Imported selenium components (webdriver, By, WebDriverWait, and expected_conditions) for web scraping.
- Imported time for adding delays.
- Initialized the Chrome WebDriver to control the browser for web scraping.
- Defined the URL of the Vivino explore page to scrape wine information.
- Created an empty list named wines to hold the scraped data.
- Created a function scrape_page() to handle the extraction of wine data from the page.
- Added a delay to ensure the page loads completely.
- Used WebDriverWait to find all wine elements on the page.
- Iterated over each wine element to extract specific details (wine name, producer, rating, country, and price).
- Added the extracted data to the wines list.
- Checked if 100 wines have been scraped to decide whether to continue or stop.
- Loaded the initial URL in the browser using driver.get().
- Called the scrape_page() function to start the scraping process.
- Used JavaScript to scroll down the page to load more results.
- Repeated the scrolling and scraping process until no new results appeared or the target number of wines was reached.
- Converted the wines list into a pandas DataFrame.
- Closed the WebDriver to release resources.

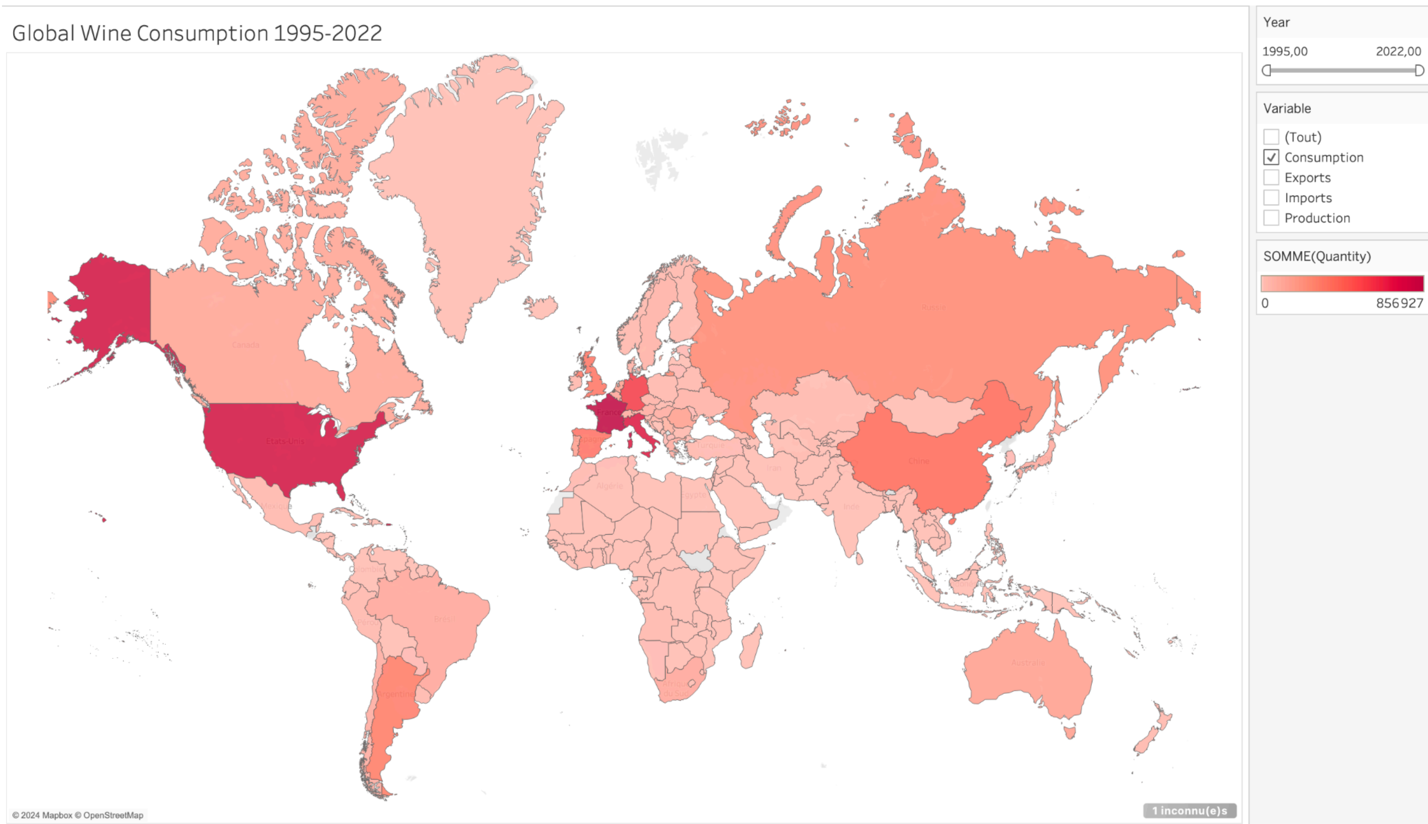
Data Cleaning

- Replaced commas in the rating column with dots and converted the values to float.
- Replaced commas in the price column with dots and converted the values to float.
- Split the country column into region and country columns.
- Save the DataFrame to a CSV file - **vivino_scraped.csv**

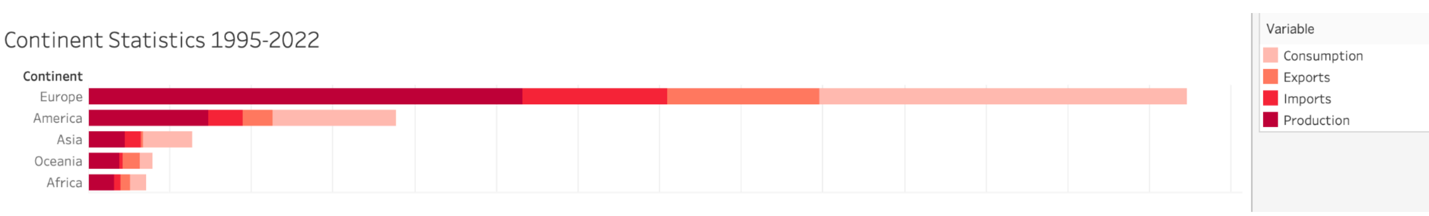
Exploratory Data Analysis and Visualizations

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

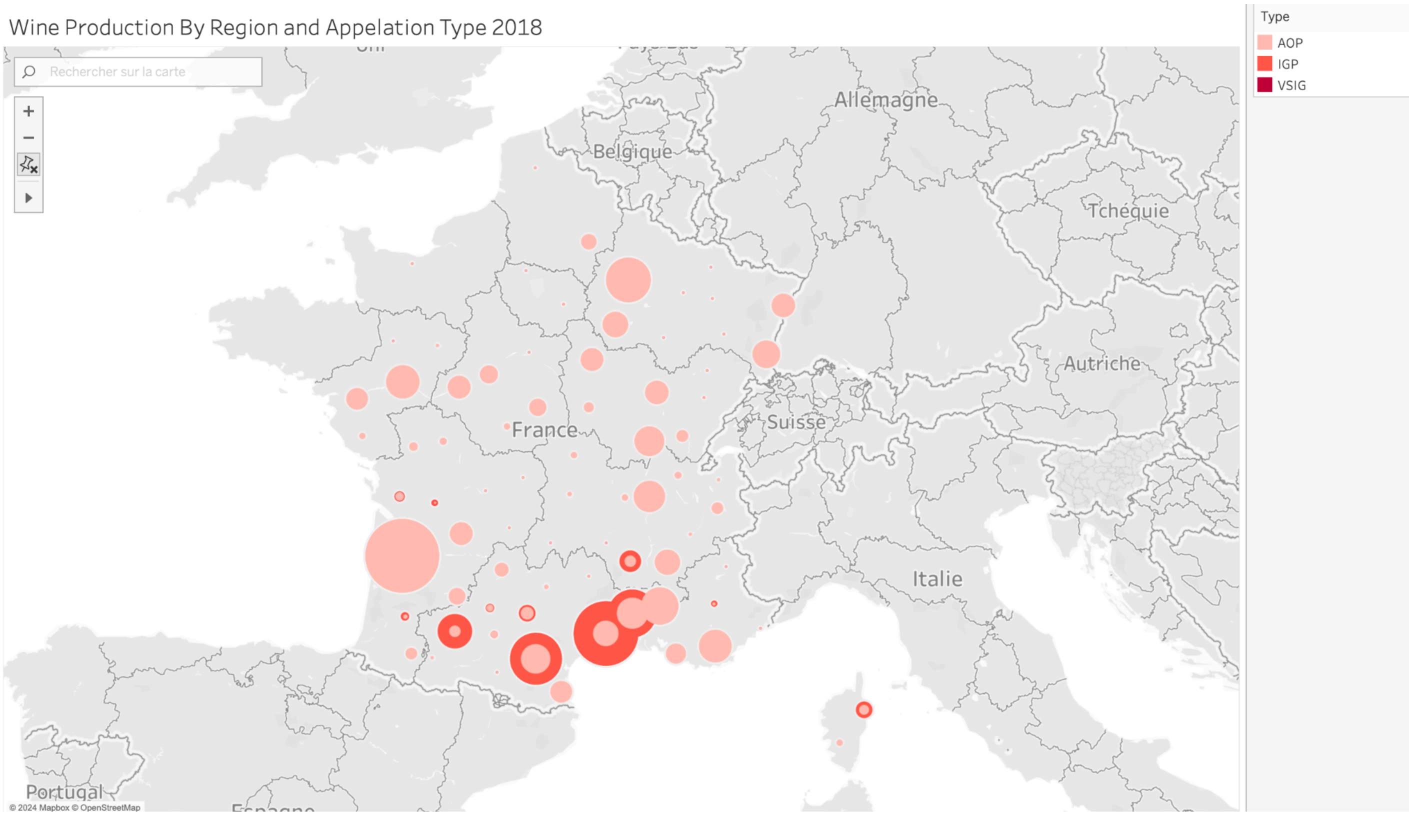
Here you can see the world map by wine consumption quantities over the past 27 years.



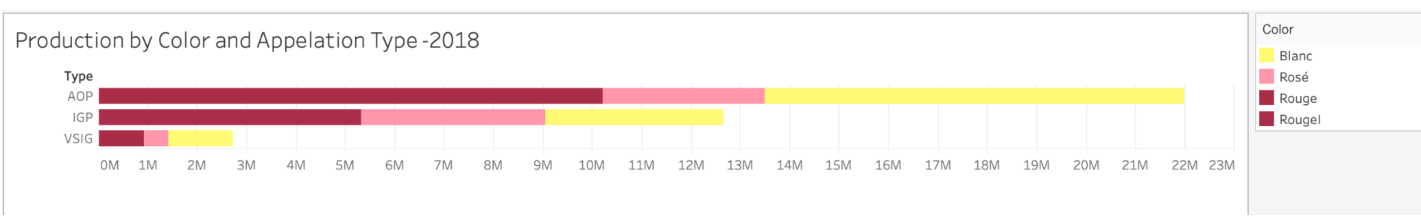
Quantities of consumption, exports, imports and production by continent.



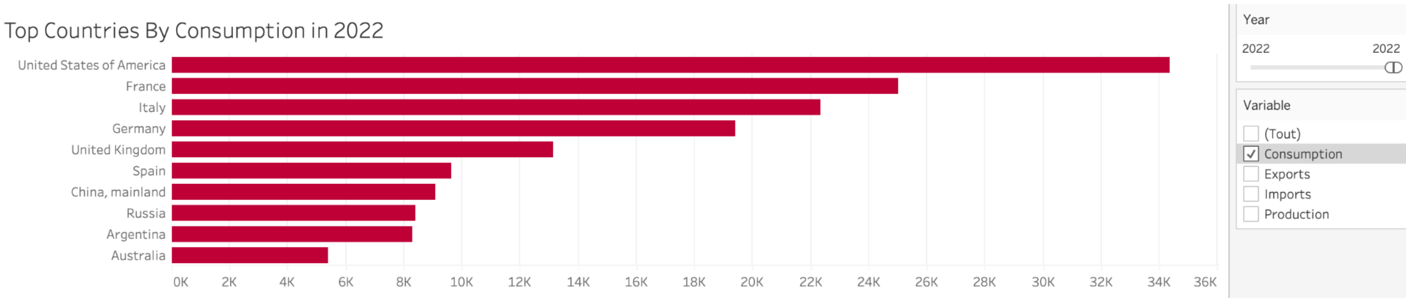
Let's focus on France. We can see the appellation types (color) and quantities of production (circle sizes) for different regions of France.

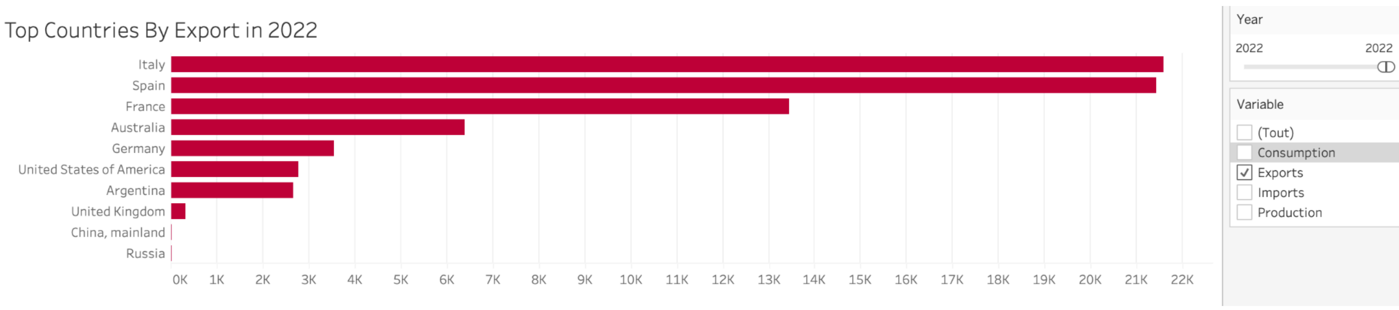


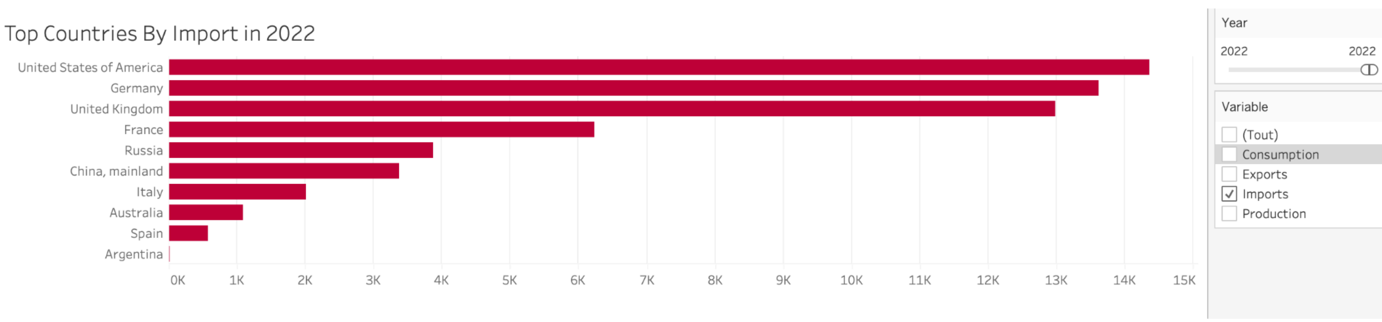
We can see what type (color) of wine is predominant by the appellation type.

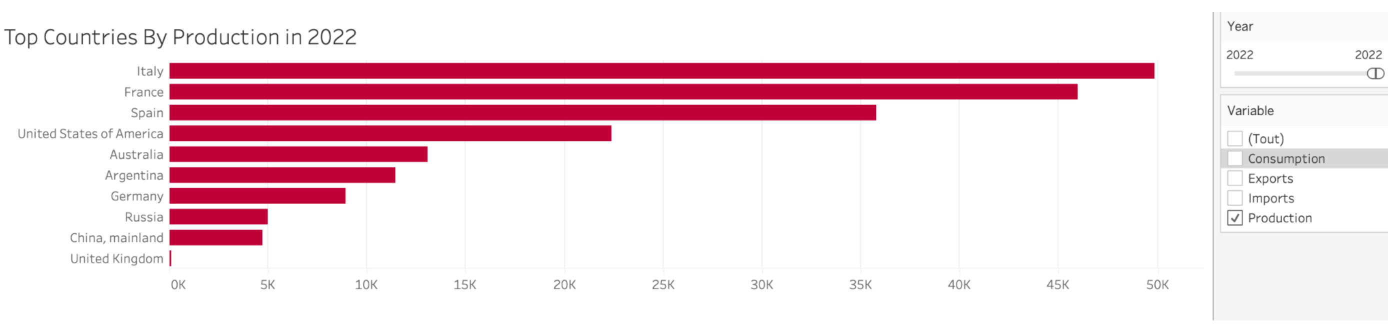


Let’s look at the top countries by consumption,export, import and production quantities for 2022.

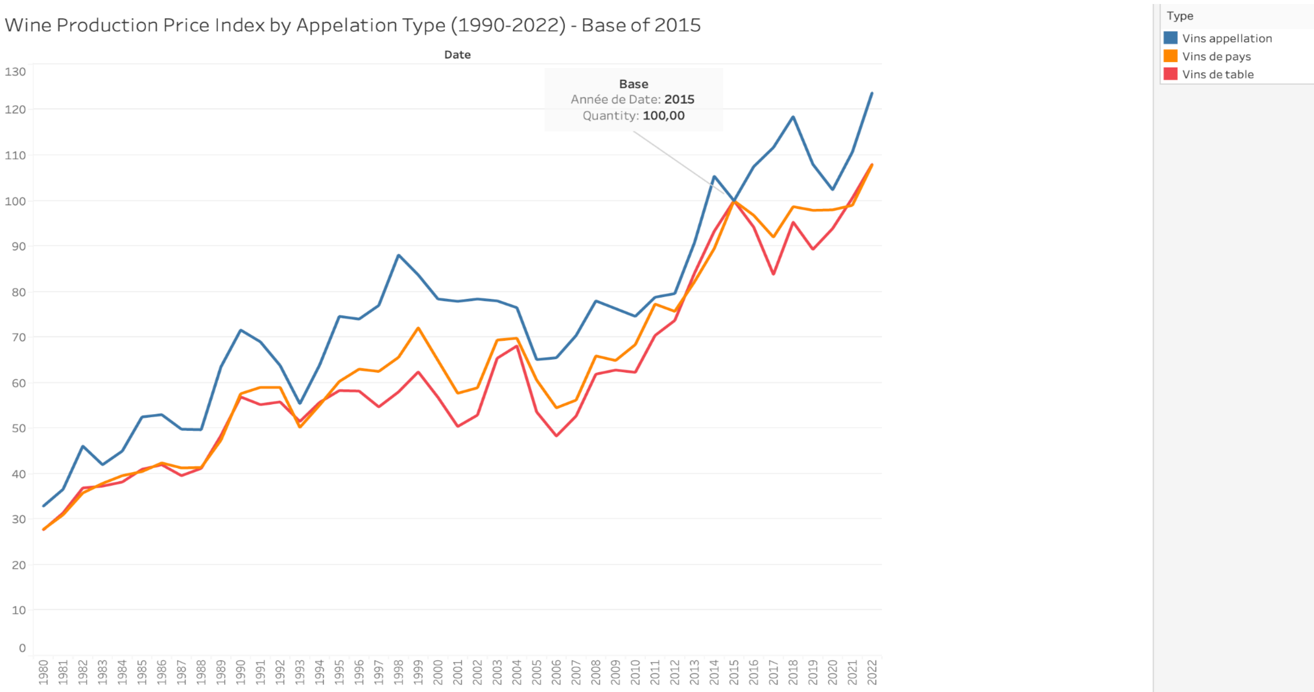




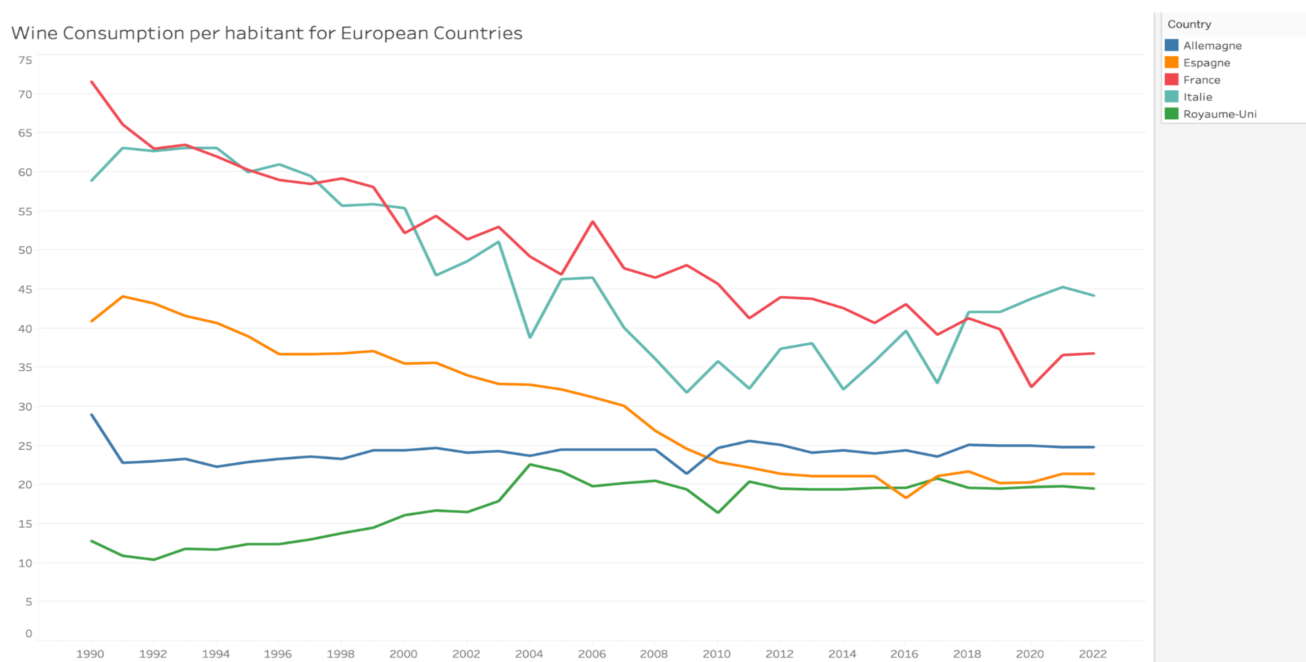




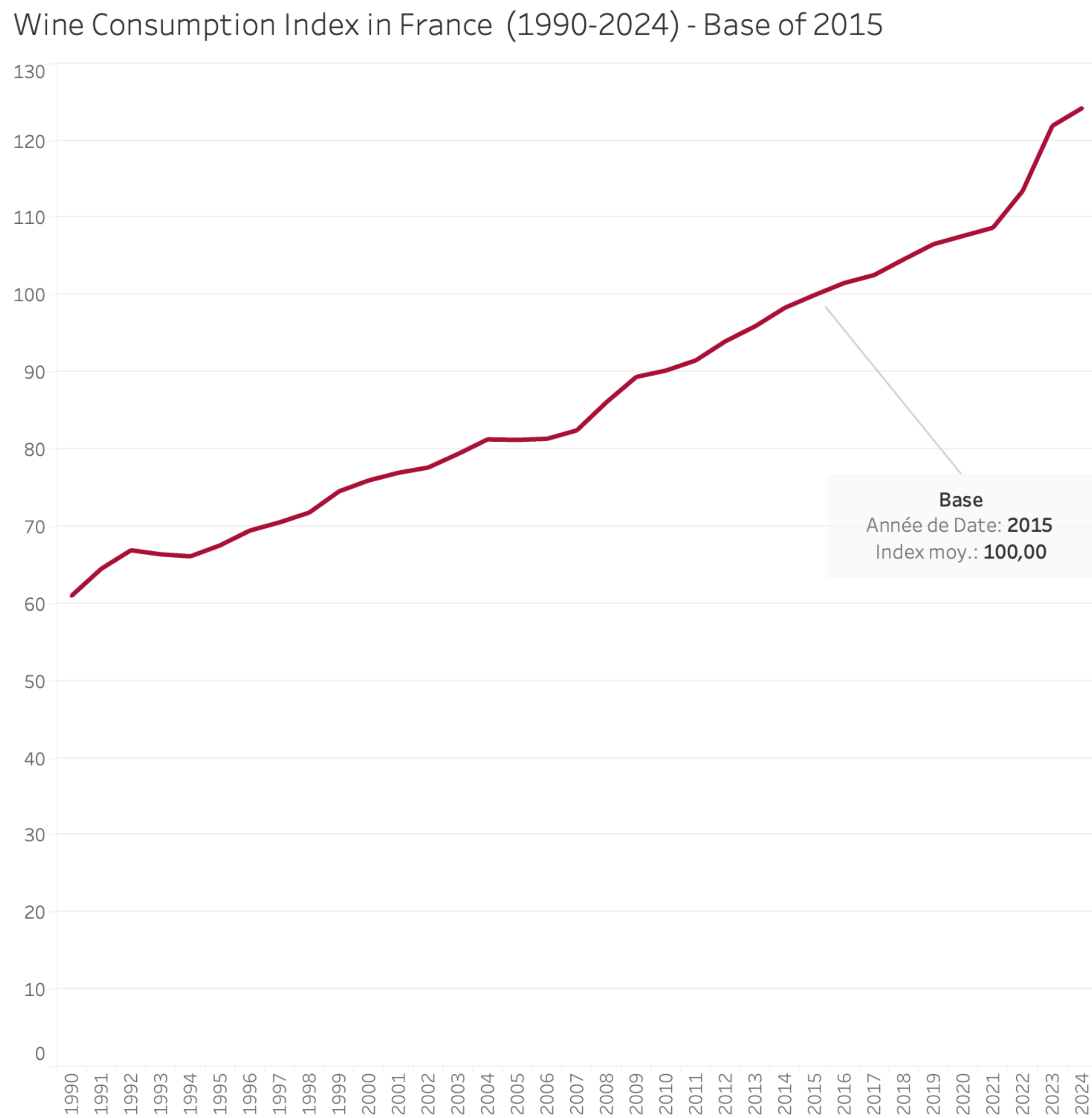
Let’s look at the wine production prices. This is the price index based on 2015 prices. We see a clear increase in price for every appellation type.



Let’s move on to the consumption part. Here we can observe wine consumption per habitant of several Euripean countries. We see that We observe that consumption in France is clearly decreasing compared to other countries.



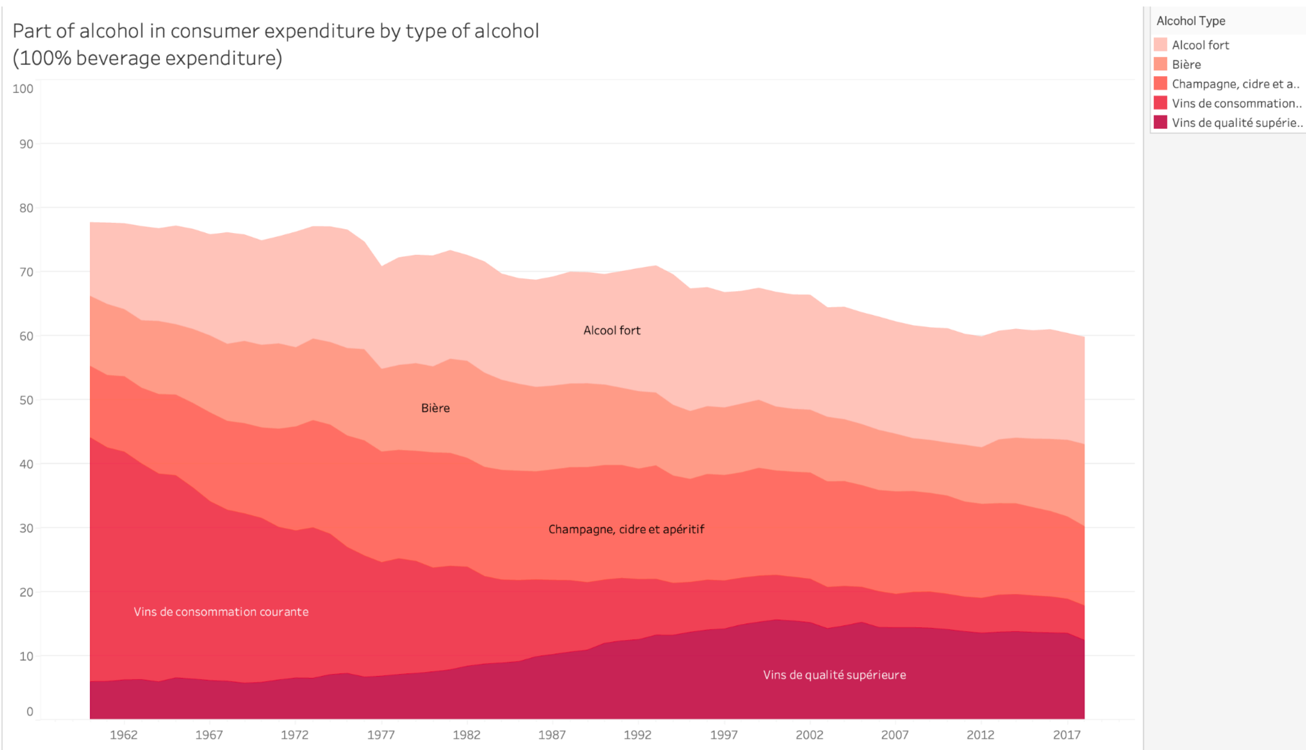
Let's look at the wine consumption prices. This is a price index based on 2015 prices. We see a clear increase in prices which might be explained by the increased price of production.



And lastly, let’s analyze the part of different alcohol types in the expenditures of french consumers.

This is a distribution of expenses for drinks in general. Non-alcoholic beverages are hidden from the table, so that we see clearly the part of alcoholic beverages and the trends associated with it.

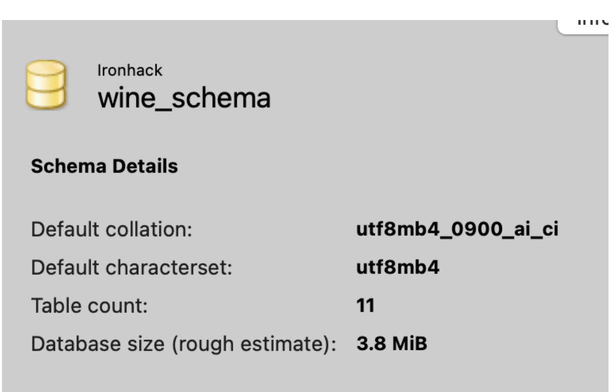
We can observe that customers shifted from consuming table wine to consuming higher quality wine. Customers are becoming more educated and appreciative of good wine.

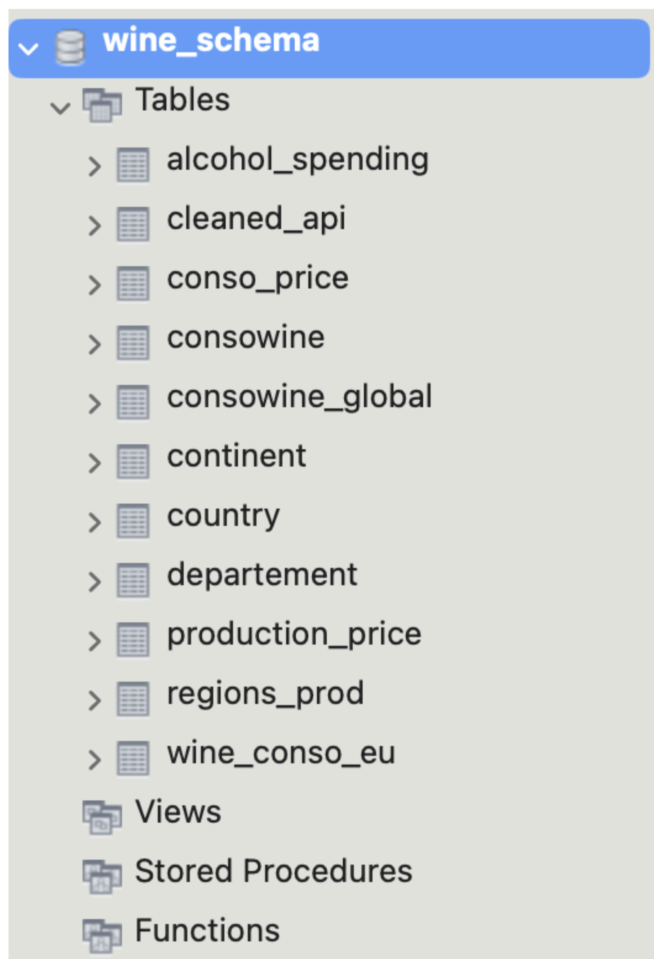


Creation of the database and ERD

When it comes to choice of database type, since I have structured data organized into interrelated tables with a predefined schema, relationships, and foreign keys it seemed appropriate to use a Relational Database. This type of database will help me to reduce data redundancy and improve data integrity, easily manipulate data, and of course, use SQL to perform queries that involve joining data from multiple tables.

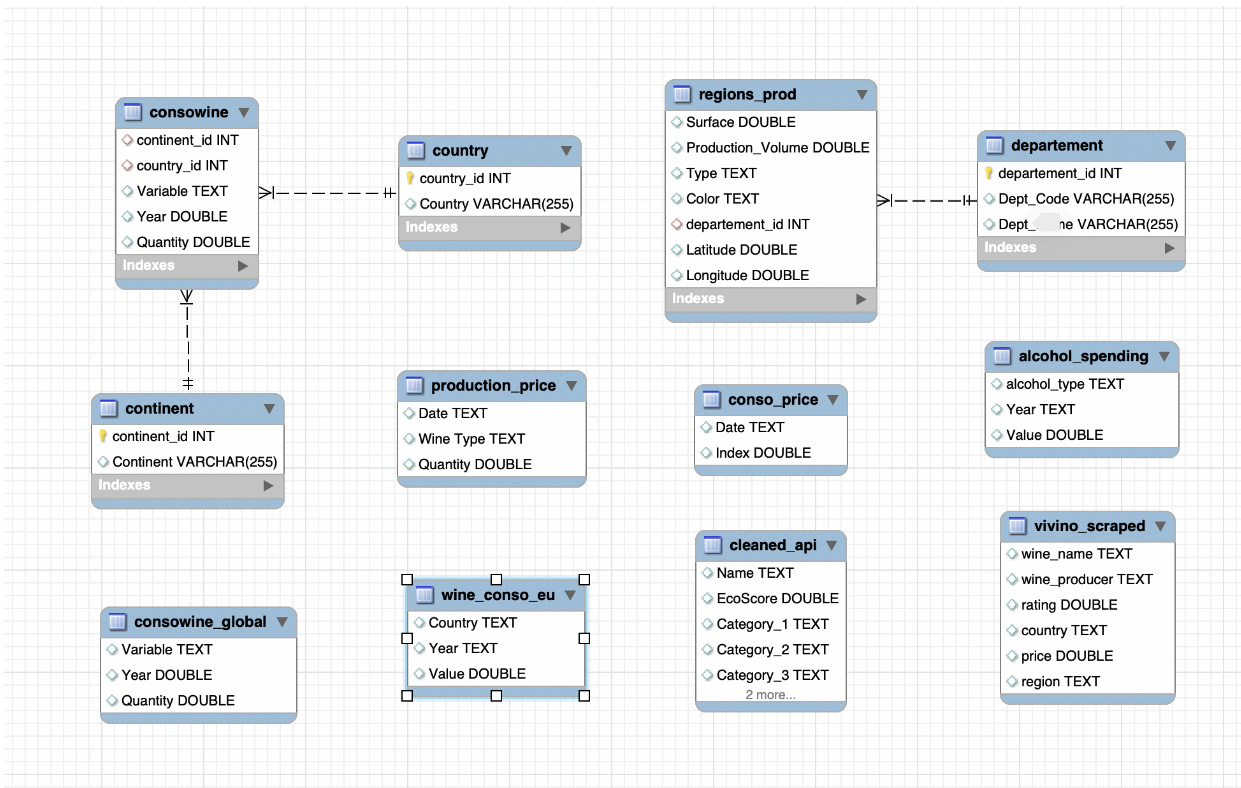
The 'wine_schema' database was created in MySQL Workbench to store11 tables related to the wine market analysis which I collected using flat files, API and web scraping:





Entities. ERD

I used a reverse engineering function to draw the ERD and show the relationships.
Country_id, continent_id and department_id are used as primary keys to connect the tables.



MySQL Queries

Let's run several queries to explore the data on mySQL.

```
-- Average yearly wine consumption per continent
SELECT
    cont.Continent,
    AVG(cw.Quantity) AS Avg_Consumption
FROM
    wine_schema.consowine AS cw
JOIN
    wine_schema.continent AS cont
ON
    cw.continent_id = cont.continent_id
WHERE
    cw.Variable = 'Consumption'
GROUP BY
    cont.Continent
ORDER BY
    Avg_Consumption DESC;
```

Continent	Avg_Consumption
Europe	3808.411815812337
America	1256.2603550295858
Asia	467.4619124797407
Oceania	344.1192468619247
Africa	133.44368600682594

```sql
-- Wine production vs consumption by country for 2020 top 10
SELECT
    prod_c.Country AS Country,
    prod.Quantity AS Production_Quantity,
    cons.Quantity AS Consumption_Quantity
FROM
    (SELECT
        country_id,
        Quantity
     FROM
        wine_schema.consowine
     WHERE
        Variable = 'Production'
        AND Year = 2020) AS prod
JOIN
    wine_schema.country AS prod_c
ON
    prod.country_id = prod_c.country_id
JOIN
    (SELECT
        country_id,
        Quantity
     FROM
        wine_schema.consowine
     WHERE
        Variable = 'Consumption'
        AND Year = 2020) AS cons
ON
    prod.country_id = cons.country_id
JOIN
    wine_schema.country AS cons_c
ON
    cons.country_id = cons_c.country_id
ORDER BY
    prod.Quantity DESC
    limit 10;
```

Country	Production_Quant...	Consumption_Quant...
Italy	49066	24200
France	46673	23202
Spain	40949	9218
United States of America	22750	32854
Australia	10900	5950
Argentina	10796	9430
South Africa	10385	2983
Chile	10337	2392
Germany	8405	19844
China, mainland	6587	12396

```sql
-- Top alcohol type by spending
SELECT
    alcohol_type AS Alcohol_Type,
    SUM(Value) AS Total_Spending
FROM
    wine_schema.alcohol_spending
GROUP BY
    alcohol_type
ORDER BY
    Total_Spending DESC
LIMIT 10;
```

Alcohol_Type	Total_Spending
Alcool fort	1002.4299999999998
Champagne, cidre et apéritif	923.4099999999997
Vins de consommation courante	856.4199999999998
Bière	683.7199999999999
Vins de qualité supérieure	624.99

```sql
-- Select top 5 producer regions by type
(
    SELECT
        d.Dept_Name AS Region,
        r.Type,
        SUM(r.Production_Volume) AS Total_Production
    FROM
        wine_schema.regions_prod AS r
    JOIN
        wine_schema.departement AS d
    ON
        r.departement_id = d.departement_id
    WHERE
        r.Type = 'AOP'
    GROUP BY
        d.Dept_Name, r.Type
    ORDER BY
        Total_Production DESC
    LIMIT 5
)
```

```sql
UNION ALL
(
    SELECT
        d.Dept_Name AS Region,
        r.Type,
        SUM(r.Production_Volume) AS Total_Production
    FROM
        wine_schema.regions_prod AS r
    JOIN
        wine_schema.departement AS d
    ON
        r.departement_id = d.departement_id
    WHERE
        r.Type = 'IGP'
    GROUP BY
        d.Dept_Name, r.Type
    ORDER BY
        Total_Production DESC
    LIMIT 5
)

UNION ALL
(
    SELECT
        d.Dept_Name AS Region,
        r.Type,
        SUM(r.Production_Volume) AS Total_Production
    FROM
        wine_schema.regions_prod AS r
    JOIN
        wine_schema.departement AS d
    ON
        r.departement_id = d.departement_id
    WHERE
        r.Type = 'VSIG'
    GROUP BY
        d.Dept_Name, r.Type
    ORDER BY
        Total_Production DESC
    LIMIT 5
);
```

Region	Type	Total_Producti...
Gironde	AOP	5041735
Marne	AOP	1829049
Vaucluse	AOP	1271676
Maine-et-Loire	AOP	1000709
Var	AOP	963810
Hérault	IGP	3847675
Aude	IGP	2474671
Gard	IGP	2103624
Gers	IGP	1075092
Vaucluse	IGP	490333
Gers	VSIG	582811
Hérault	VSIG	459812
Gard	VSIG	254943
Aude	VSIG	241779
Loire-Atlantiq…	VSIG	239219

```sql
-- Calculate the total production of wine for each country over the last 20 years and show the top 10
SELECT
    c.country_id,
    ct.Country,
    SUM(c.Production) AS Total_Production_Last_20_Years
FROM
    (SELECT country_id, Year, Quantity AS Production
     FROM wine_schema.consowine
     WHERE Variable = 'Production' AND Year >= YEAR(CURDATE()) - 20) AS c
JOIN
    wine_schema.country AS ct ON c.country_id = ct.country_id
GROUP BY
    c.country_id, ct.Country
ORDER BY
    Total_Production_Last_20_Years DESC
LIMIT 10;
```

country_id	Country	Total_Production_Last_20_Ye...
97	Italy	961137
71	France	921036
188	Spain	739630
210	United States of America	454519
8	Argentina	267334
11	Australia	247969
43	China, mainland	218660
40	Chile	208801
186	South Africa	201724
77	Germany	178386

```sql
-- Find the most frequent occurrences in the Category_5 column from API
SELECT
    Category_5,
    COUNT(*) AS Occurrence
FROM
    wine_schema.cleaned_api
GROUP BY
    Category_5
ORDER BY
    Occurrence DESC
LIMIT 10;
```

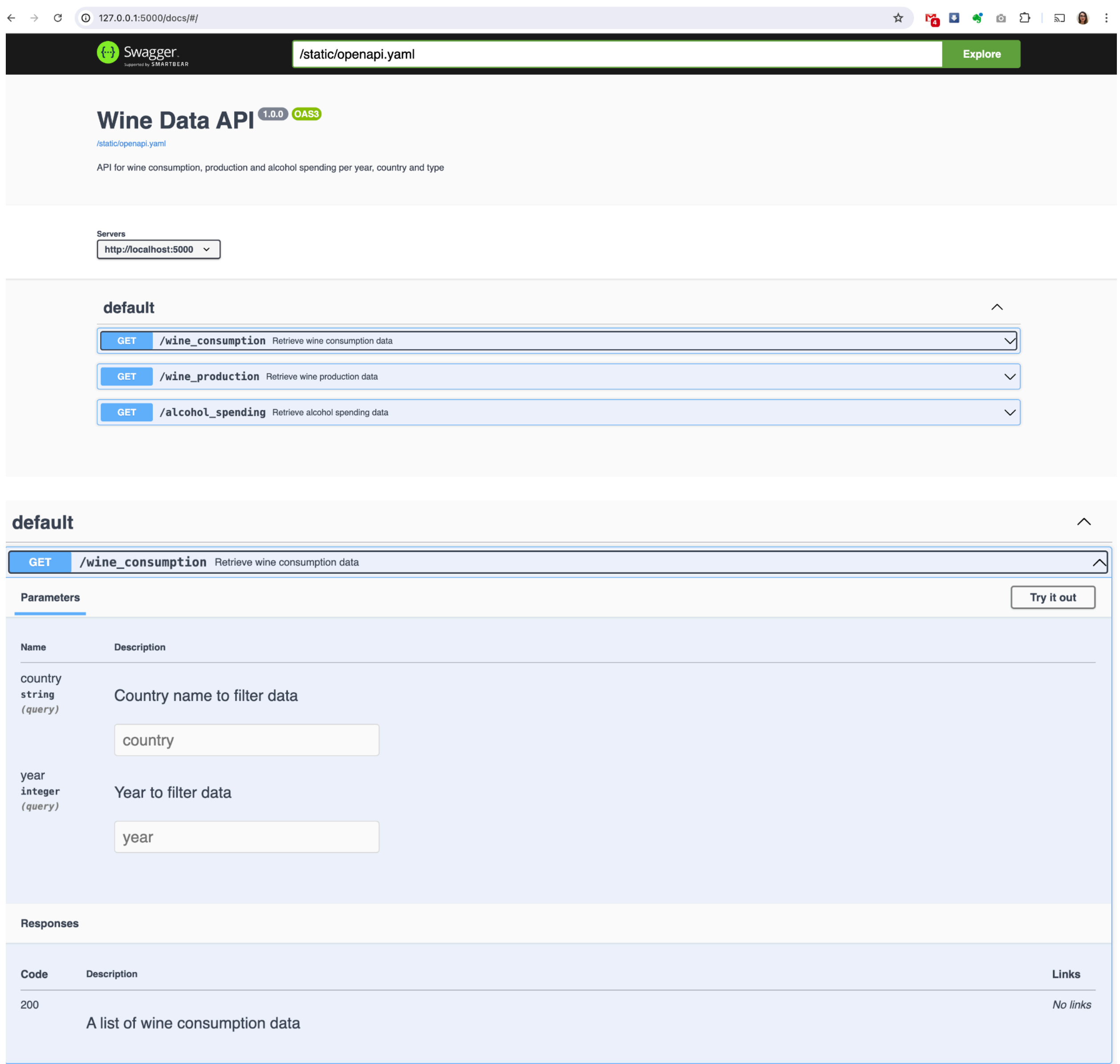
Category_5	Occurrence
NULL	896
Vins rouges	330
Vins blancs	146
Bordeaux	142
Vins rosés	125
Champagnes	112
Côtes du Rhône	97
Bourgogne	88
Vins effervescents	87
Vins doux	59

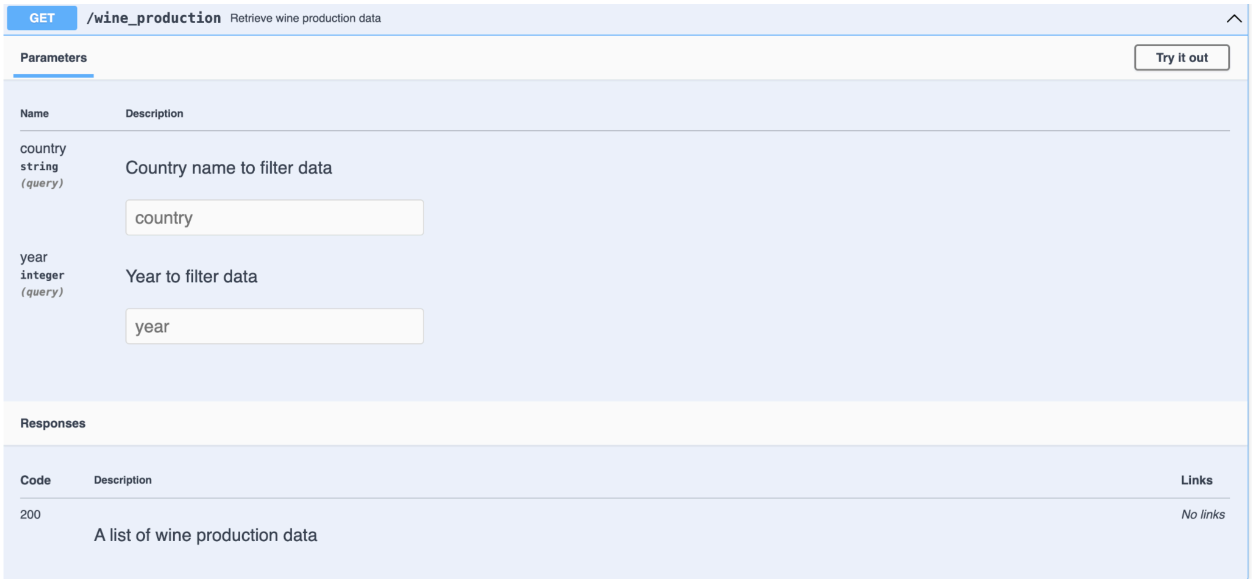
My API Creation

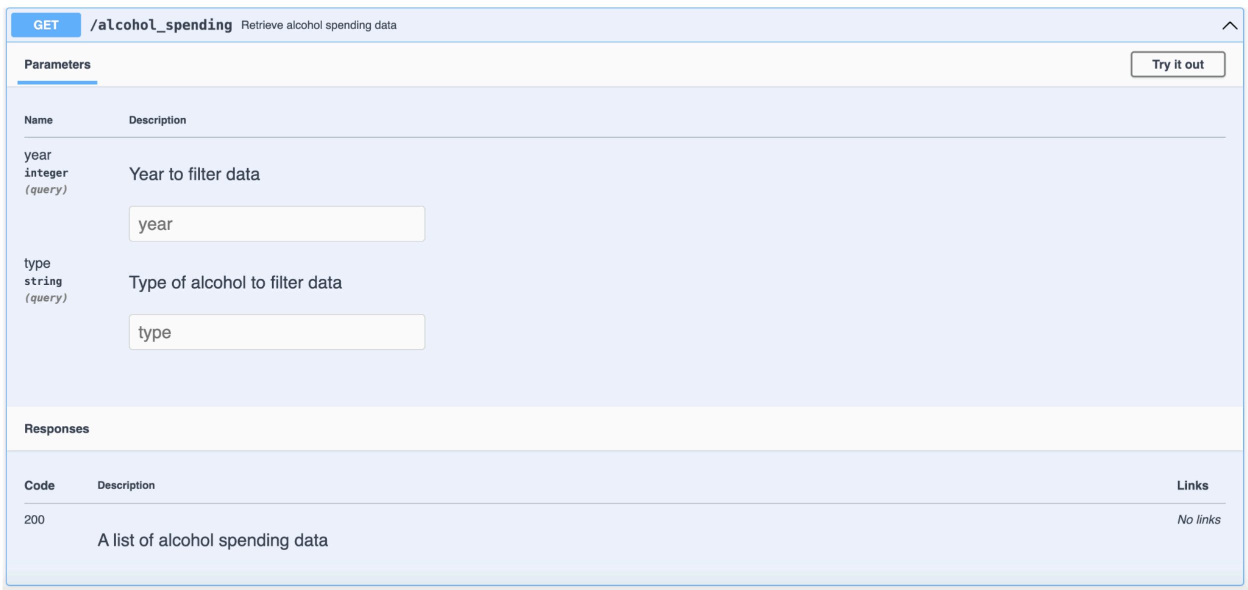
I created a Restful API to display the data. Steps I followed:

- Import essential libraries such as Flask for creating the web application, pymysql for database connections, and additional libraries for handling requests and responses.
- Create an instance of the Flask application which serves as the main entry point for the web server.
- Set up Swagger UI to provide interactive API documentation. The Swagger UI blueprint is registered with the Flask app, and an endpoint is created to serve the OpenAPI specification file.
- Define a function to establish and return a connection to the MySQL database using pymysql. The function uses environment variables for database credentials for security.
- Create an endpoint **/wine_consumption** that accepts GET requests. This endpoint retrieves wine consumption data from the database. Query parameters for country and year are used to filter results. SQL queries are dynamically constructed based on the provided parameters, and results are returned in JSON format.
- Create an endpoint **/wine_production** that accepts GET requests. This endpoint retrieves wine production data from the database. Similar to the wine consumption endpoint, query parameters for country and year are used to filter results, and results are returned in JSON format.
- Create an endpoint **/alcohol_spending** that accepts GET requests. This endpoint retrieves data on alcohol spending from the database. Query parameters for year and type are used to filter results. The query is dynamically constructed based on the provided parameters, and the results are returned in JSON format.
- The Flask application is run with debugging enabled to facilitate development and troubleshooting.

Let’s look at Swagger API documentation for my API.

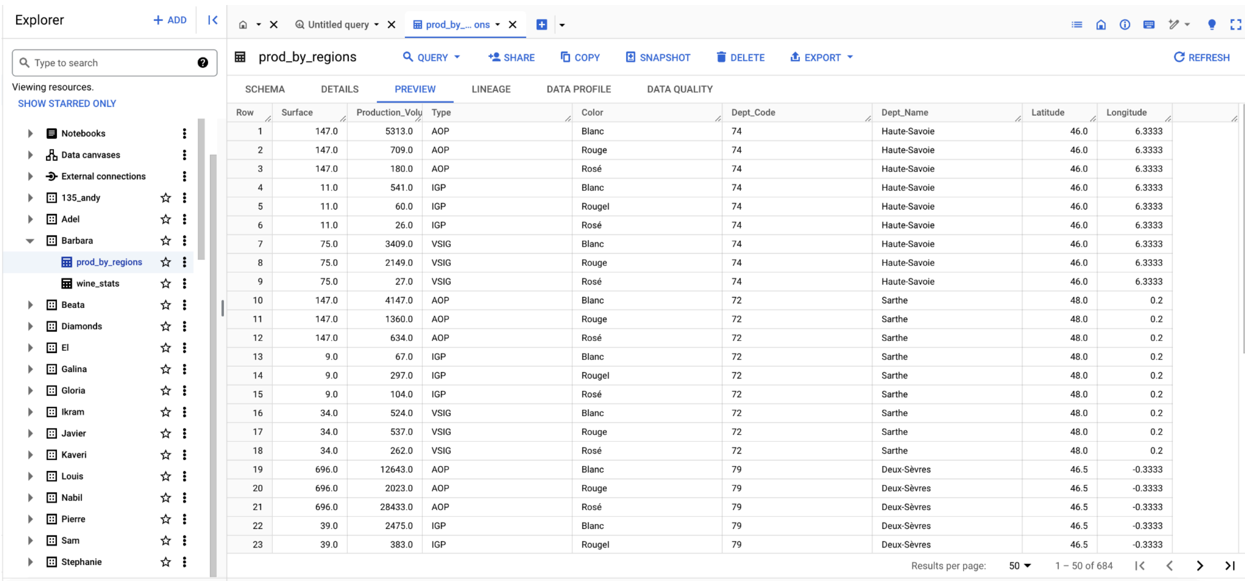



GET
/wine_production
Retrieve wine production data
Parameters
Try it out
Name
Description
country
string
(query)
Country name to filter data
country
year
integer
(query)
Year to filter data
year
Responses
Code
Description
Links
200
A list of wine production data
No links

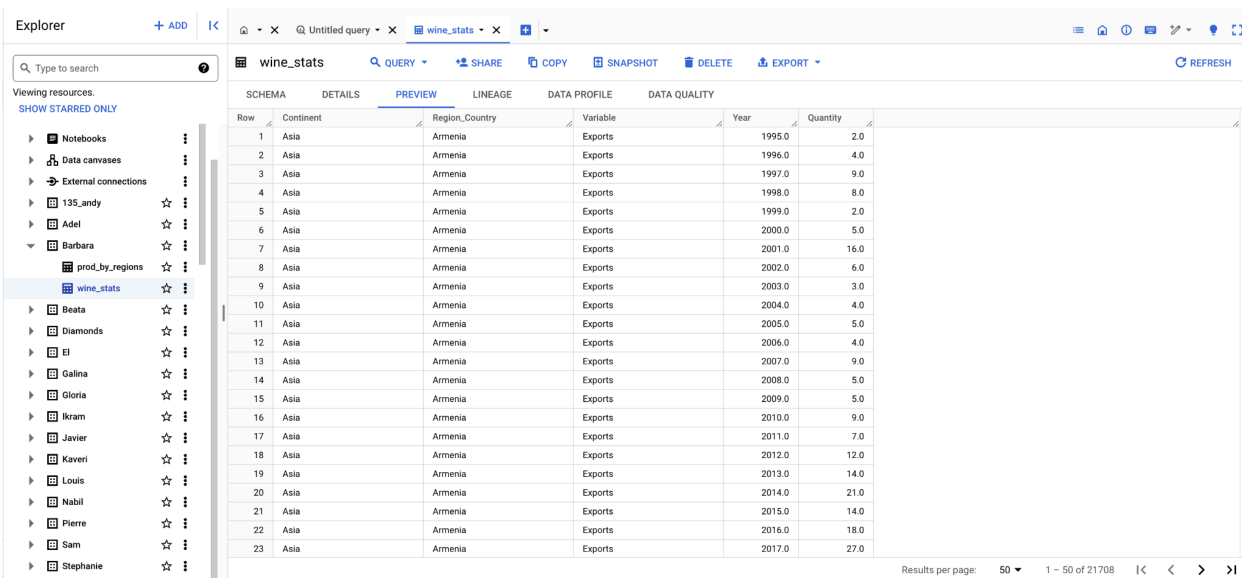

GET
/alcohol_spending
Retrieve alcohol spending data
Parameters
Try it out
Name
Description
year
integer
(query)
Year to filter data
year
type
string
(query)
Type of alcohol to filter data
type
Responses
Code
Description
Links
200
A list of alcohol spending data
No links

Bigquery

In my project, I created two new denormalized view tables in BigQuery to streamline data analysis and querying processes. Initially, I considered partitioning these tables by country to enhance query performance. However, given that the dataset comprises only 20,000 rows, partitioning was deemed unnecessary. Below are the previews of the two tables:



Row	Surface	Production_Volu	Type	Color	Dept_Code	Dept_Name	Latitude	Longitude
1	147.0	5313.0	AOP	Blanc	74	Haute-Savoie	46.0	6.3333
2	147.0	709.0	AOP	Rouge	74	Haute-Savoie	46.0	6.3333
3	147.0	180.0	AOP	Rosé	74	Haute-Savoie	46.0	6.3333
4	11.0	541.0	IGP	Blanc	74	Haute-Savoie	46.0	6.3333
5	11.0	60.0	IGP	Rougel	74	Haute-Savoie	46.0	6.3333
6	11.0	26.0	IGP	Rosé	74	Haute-Savoie	46.0	6.3333
7	75.0	3409.0	VSIG	Blanc	74	Haute-Savoie	46.0	6.3333
8	75.0	2149.0	VSIG	Rouge	74	Haute-Savoie	46.0	6.3333
9	75.0	27.0	VSIG	Rosé	74	Haute-Savoie	46.0	6.3333
10	147.0	4147.0	AOP	Blanc	72	Sarthe	48.0	0.2
11	147.0	1360.0	AOP	Rouge	72	Sarthe	48.0	0.2
12	147.0	634.0	AOP	Rosé	72	Sarthe	48.0	0.2
13	9.0	67.0	IGP	Blanc	72	Sarthe	48.0	0.2
14	9.0	297.0	IGP	Rougel	72	Sarthe	48.0	0.2
15	9.0	104.0	IGP	Rosé	72	Sarthe	48.0	0.2
16	34.0	524.0	VSIG	Blanc	72	Sarthe	48.0	0.2
17	34.0	537.0	VSIG	Rouge	72	Sarthe	48.0	0.2
18	34.0	262.0	VSIG	Rosé	72	Sarthe	48.0	0.2
19	696.0	12643.0	AOP	Blanc	79	Deux-Sèvres	46.5	-0.3333
20	696.0	2023.0	AOP	Rouge	79	Deux-Sèvres	46.5	-0.3333
21	696.0	28433.0	AOP	Rosé	79	Deux-Sèvres	46.5	-0.3333
22	39.0	2475.0	IGP	Blanc	79	Deux-Sèvres	46.5	-0.3333
23	39.0	383.0	IGP	Rougel	79	Deux-Sèvres	46.5	-0.3333



Row	Continent	Region_Country	Variable	Year	Quantity
1	Asia	Armenia	Exports	1995.0	2.0
2	Asia	Armenia	Exports	1996.0	4.0
3	Asia	Armenia	Exports	1997.0	9.0
4	Asia	Armenia	Exports	1998.0	8.0
5	Asia	Armenia	Exports	1999.0	2.0
6	Asia	Armenia	Exports	2000.0	5.0
7	Asia	Armenia	Exports	2001.0	16.0
8	Asia	Armenia	Exports	2002.0	6.0
9	Asia	Armenia	Exports	2003.0	3.0
10	Asia	Armenia	Exports	2004.0	4.0
11	Asia	Armenia	Exports	2005.0	5.0
12	Asia	Armenia	Exports	2006.0	4.0
13	Asia	Armenia	Exports	2007.0	9.0
14	Asia	Armenia	Exports	2008.0	5.0
15	Asia	Armenia	Exports	2009.0	5.0
16	Asia	Armenia	Exports	2010.0	9.0
17	Asia	Armenia	Exports	2011.0	7.0
18	Asia	Armenia	Exports	2012.0	12.0
19	Asia	Armenia	Exports	2013.0	14.0
20	Asia	Armenia	Exports	2014.0	21.0
21	Asia	Armenia	Exports	2015.0	14.0
22	Asia	Armenia	Exports	2016.0	18.0
23	Asia	Armenia	Exports	2017.0	27.0

Machine Learning (in progress)

Overview

This project aims to develop a Streamlit application that integrates a food recognition system with a wine recommendation engine. By leveraging machine learning and deep learning techniques, the application provides users with personalized wine recommendations based on images of their meals. The project uses the FOOD 101 image database to train a convolutional neural network (CNN) model for accurate food recognition and incorporates various data sources for the wine recommendation component.

Food Recognition System

Data Preparation:

- Utilized the FOOD 101 image database, which contains a diverse set of food images across 101 categories.
- Prepared the data by reading image paths and labels from the dataset's metadata files.
- Sampled a subset of 5,000 images for training and 1,000 images for testing to ensure manageable computation and quick iterations.

Image Preprocessing:

- Developed a helper function to load and preprocess images, including resizing and normalization to improve model training efficiency.
- Converted image paths and labels into TensorFlow datasets, applying the preprocessing function to each image.

Model Architecture:

- Designed a deep learning model using the Sequential API from TensorFlow and Keras.
- Implemented a convolutional neural network (CNN) with multiple layers:
    - Convolutional layers with ReLU activation functions for feature extraction.

- MaxPooling layers to reduce spatial dimensions and computational load.
- Flattening layer to convert the 2D matrix data to a 1D vector.
- Dense layers for classification, with the final layer using softmax activation to output probabilities for each food category.

Training and Evaluation:

- Compiled the model with the Adam optimizer and sparse categorical crossentropy loss function, specifying accuracy as the evaluation metric.
- Implemented early stopping and model checkpoint callbacks to prevent overfitting and save the best model during training.
- Trained the model for up to 50 epochs, monitoring validation performance to ensure robust learning.
- Saved the trained model in the Keras format for easy deployment and integration into the application.

Streamlit Application

- Designed a user-friendly interface using Streamlit, allowing users to upload meal images and receive wine recommendations.
- Integrated the trained food recognition model to classify uploaded images in real-time.

GDPR

After a thorough assessment of the data collected for this project, I can confidently affirm that no personal data has been utilized in any part of the processes. The sources of the data are entirely public and at a country level, ensuring full transparency and accessibility. Therefore, this project adheres to the guidelines and principles of the General Data Protection Regulation (GDPR).

Conclusion

The project provided me with deep insights into the wine market, both globally and within France. Key conclusions from this analysis include:

Regional Specialization: France continues to maintain its reputation for diverse and high-quality wine production, with specific regions excelling in different types of wine.

Market Trends: Both production costs and consumer prices for wine are on the rise, indicating economic pressures and market dynamics that affect the entire supply chain.

Leading countries: France remains top positions in consumption, export, import and production of wine

Global Standing: French wines remain highly regarded globally, as evidenced by their prominence in the top-rated wines on platforms like Vivino.

Consumption Patterns: There is a noticeable shift in consumption patterns, with French consumers increasingly favoring higher-quality wines over table wines.

Given these insights, it is clear that there is a strong foundation for developing a wine recommendation application. The application can leverage the detailed data analysis and trends identified in this project to enhance the wine experience for users.